JP03/13405

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　９月１７日

出願番号
Application Number: 特願２００３－３２４８０５

[ST. 10/C]: [ＪＰ２００３－３２４８０５]

出願人
Applicant(s): 松下電器産業株式会社

２００３年１１月２１日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫

出証番号　出証特２００３－３０９６４４８

【書類名】　特許願
【整理番号】　2931050083
【提出日】　平成15年 9月17日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 9/06
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　里　雄二
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　山口　孝雄
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　佐藤　潤一
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　武井　一朗
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　伊藤　智祥
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100105050
　【弁理士】
　【氏名又は名称】　鷲田　公一
【先の出願に基づく優先権主張】
　【出願番号】　特願2002-311815
　【出願日】　平成14年10月25日
【先の出願に基づく優先権主張】
　【出願番号】　特願2003-133566
　【出願日】　平成15年 5月12日
【手数料の表示】
　【予納台帳番号】　041243
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9700376

【書類名】特許請求の範囲
【請求項１】
プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入する透かし情報挿入手段と、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させない透かし検証コードを前記プログラムに挿入するコード挿入手段と、を具備し、
前記透かし検証コードを前記配布先に関わらず同じにしたことを特徴とする透かし挿入装置。
【請求項２】
前記透かし情報を、プログラムの配布先を一意に決定するＩＤ情報から生成することを特徴とする請求項１記載の透かし挿入装置。
【請求項３】
前記透かし情報から所定の定数を出力する関数を定義し、前記関数を変数に代入する式を前記プログラムに挿入する関数挿入手段を具備し、
前記透かし検証コードは、前記変数と前記定数が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐であり、
前記定数を前記配布先に関わらず同じにしていることを特徴とする請求項１または請求項２記載の透かし挿入装置。
【請求項４】
前記透かし検証コードは、前記プログラムを正しく動作させるのに必要なものであることを特徴とする請求項１または請求項２記載の透かし挿入装置。
【請求項５】
前記透かし検証コードは、前記プログラムから取り出した判定分岐に前記透かし情報から生成した前記判定分岐の判定文に影響を与えない計算式を挿入したものであることを特徴とする請求項４記載の透かし挿入装置。
【請求項６】
請求項１から請求項５のいずれかに記載の透かし挿入装置が前記透かし情報および前記透かし検証コードを挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出し、前記透かし情報に基づいて前記配布先を一意に特定するＩＤ情報を生成する透かし検出手段と、を具備し、
生成した前記ＩＤ情報に基づき配布先を特定することを特徴とする透かし取出装置。
【請求項７】
請求項１から請求項５のいずれかに記載の透かし挿入装置と、請求項６に記載の透かし取出装置と、を具備したことを特徴とするプログラム不正配布防止システム。
【請求項８】
前記透かし挿入装置を前記配布先に設けたことを特徴とする請求項７記載のプログラム不正配布防止システム。
【請求項９】
配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記プログラムに挿入するステップと、を具備したことを特徴とする透かし挿入方法。
【請求項１０】
プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報の挿入箇所の周辺もしくは前記プログラムの全体を、仕様を維持したまま変換するステップと、を有することを特徴とする透かし挿入方法。
【請求項１１】
コンピュータに、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記配布用プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記配布用プログラムに挿入するステップと、を行わせること

を特徴とする透かし挿入プログラム。
【請求項１２】
プログラムの配布先ごとに異なる透かし情報をプログラムに挿入する透かし挿入手段と、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を、維持したまま変換する変換手段と、を具備したことを特徴とする透かし挿入装置。
【請求項１３】
前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えない実行コードの組を挿入することを特徴とする請求項１２記載の透かし挿入装置。
【請求項１４】
前記透かし情報の挿入箇所を示す識別情報を記憶することを特徴とする請求項１２または請求項１３に記載の透かし挿入装置。
【請求項１５】
前記識別情報は、メソッド名もしくは行番号であることを特徴とする請求項１４に記載の透かし挿入装置。
【請求項１６】
前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えないように難読化処理することを特徴とする請求項１２記載の透かし挿入装置。
【請求項１７】
請求項１２から請求項１６のいずれかに記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出す透かし検出手段と、を具備し、
取り出した前記透かし情報に基づき配布先を特定することを特徴とする透かし取出装置。
【請求項１８】
請求項１５または請求項１６に記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記識別情報を取得し、前記識別情報から透かし挿入箇所を特定し、特定した前記透かし挿入個所のみから前記透かし情報を取り出す透かし検出手段と、を具備し、
取り出した前記透かし情報に基づき配布先を特定することを特徴とする透かし取出装置。
【請求項１９】
コンピュータに、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を変更することなく変換するステップと、行わせることを特徴とするプログラム。
【請求項２０】
前記変換手段は、前記透かし情報を挿入する箇所以外の部分であって、順序を入れ替えても仕様に影響を与えない部分の順序を変換することを特徴とする請求項１２記載の透かし挿入装置。
【請求項２１】
前記仕様に影響を与えない部分を変換した履歴情報を保持し、前記履歴情報を用いて、前記仕様に影響を与えない部分の変換を配布先ごとに異なるようにすることを特徴とする請求項２０記載の透かし挿入装置。

【書類名】明細書
【発明の名称】透かし挿入装置および透かし取出装置
【技術分野】
【０００１】
本発明は、プログラムの不正な使用および配布を防止および抑止するためのプログラムへの透かしの挿入装置および透かし取出装置に関するものである。
【背景技術】
【０００２】
コンピュータネットワークの進展に伴い、ネットワークを介したコンピュータプログラムの流通が一般的になっている。コンピュータプログラムは、容易に複製を作成できるため、プログラムの複製が不正に２次配布されたり、プログラム中のアルゴリズムを盗用、改ざんされたりする可能性がある。したがって、このような不正利用からプログラムを保護する必要がある。
【０００３】
従来のプログラム保護の技術の一つとして、プログラムへ電子透かしを挿入する方法が挙げられる。この方法では、配布先ごとに異なる透かし情報をプログラムに埋め込んで配布する。そして、不正利用が発生した場合に、不正利用者のプログラムから透かし情報を取り出し、透かし情報を解析する。これにより、流出元を容易に検出することが可能となる。
【０００４】
具体的な透かしの挿入方法としては、たとえば、特許文献１に開示されたものがある。
【０００５】
この方法は、まず、実行順序に依存関係のないコードを検出する。次に、検出部分にダミー変数の演算を挿入する。そして、ダミー変数の演算を含む検出部分の実行順序をランダムに入れ替える。
【０００６】
このような処理を行うことにより、その実行順序を電子透かし情報として配布先ごとに変更する仕組みを実現している。
【特許文献１】特開２０００－７６０６４号公報（第３－４頁、第２図、第７図）
【発明の開示】
【発明が解決しようとする課題】
【０００７】
しかしながら、従来のプログラムへの電子透かし挿入方式は、差分攻撃に基づく透かしの改変、削除が容易であるという問題がある。
【０００８】
差分攻撃とは、複数の電子透かしの挿入されたプログラムの差分をとることで、透かしデータの挿入箇所を特定する攻撃法である。
【０００９】
従来の方式を用いて、プログラムに配布先ごとに異なる透かし情報を挿入した場合、各配布先に配布されたプログラムの間で差分をとると、透かしの挿入された個所だけが差分として浮かび上がってしまう。このように、差分攻撃で透かしの挿入位置が簡単に特定されてしまい、透かし情報の削除、改ざんが容易に可能であるという問題がある。
【００１０】
本発明は、かかる点に鑑みてなされたものであり、透かしの挿入箇所を特定されないように透かしを挿入することにより、透かしがなく、かつ正常に動作するプログラムを容易に生成できないようにすることを目的とする。
【課題を解決するための手段】
【００１１】
上記の課題を解決するための本発明は、プログラムの配布先を一意に特定するＩＤ情報から透かし情報を生成し、生成した透かし情報をプログラムに挿入し、透かし情報が改ざ

んされた場合にはプログラムを正しく動作させないものであり、かつ配布先に関わらず同じ透かし検証コードをプログラムに挿入するようにした。

【発明の効果】

【００１２】

これにより、差分攻撃により透かしである透かし検証コードが検出されないようにできる。この結果、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【００１３】

また、透かし情報を挿入したあとに、透かし挿入箇所周辺や、プログラム全体をプログラムの仕様が変更しない範囲で配布先ごとに異なるように変換するようにした。

【００１４】

これにより、プログラムの差分を取った際に、透かし情報以外の部分が差分として浮かび上がるため、透かし挿入箇所を容易に特定することができなくなる。

【発明を実施するための最良の形態】

【００１５】

本発明の第１の態様にかかる透かし挿入装置は、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入する透かし情報挿入手段と、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させない透かし検証コードを前記プログラムに挿入するコード挿入手段と、を具備し、前記透かし検証コードを前記配布先に関わらず同じにしたものである。

【００１６】

このように透かし検証コードを配布先に関わらず同じにすることにより、差分攻撃により、透かし検証コードが差分として検出されないようにできる。これにより、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、透かし検証コードを改変、削除できなくなる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【００１７】

本発明の第２の態様は、第１の態様にかかる透かし挿入装置において、配布先を一意に決定するＩＤ情報に基づき透かし情報を生成する。

【００１８】

これにより、配布先が不正配布を行った際に、配布先を一意に特定することができ、プログラムの不正流通を防止できる。

【００１９】

本発明の第３の態様は、第１の態様または第２の態様にかかる透かし挿入装置において、前記透かし情報から所定の定数を出力する関数を定義し、前記関数を変数に代入する式を前記プログラムに挿入する関数挿入手段を具備し、前記透かし検証コードは、前記変数と前記定数が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐であり、前記定数を前記配布先に関わらず同じにしている。

【００２０】

このような条件分布は、差分攻撃により検出されないので、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを改変、削除できなくなる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。

【００２１】

本発明の第４の態様は、第１の態様または第２の態様にかかる透かし挿入装置において、前記透かし検証コードは、前記プログラムを正しく動作させるのに必要なものである。

【００２２】

これにより、差分攻撃により検出した透かし情報から透かし検証コードを検出し、透かし検証コードを削除、改変してしまうとプログラムが正常に動作しないようになる。つまり、正常に動作する透かし情報のない（もしくは改変された）プログラムを生成すること

を不可能にすることができるので、プログラムの不正配布を防止できる。

【００２３】

本発明の第５の態様は、第４の態様にかかる透かし挿入装置において、前記透かし検証コードは、前記プログラムから取り出した判定分岐に前記透かし情報から生成した前記判定分岐の判定文に影響を与えない計算式を挿入したものである。

【００２４】

これにより、プログラムを正しく動作させるのに必要な透かし検証コードを入力できる。

【００２５】

本発明の第６の態様にかかる透かし取出装置は、第１の態様から第５の態様のいずれかに記載の透かし挿入装置が前記透かし情報および前記透かし検証コードを挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出し、前記透かし情報に基づいて配布先を一意に特定するＩＤ情報を生成する透かし検出手段と、を具備し、生成した前記ＩＤ情報に基づき前記配布先を特定するものである。

【００２６】

これにより、不正にプログラムを流通した配布先を特定することができる。

【００２７】

本発明の第７の態様にかかるプログラム不正配布防止システムは、第１の態様から第５の態様のいずれかに記載の透かし挿入装置と、第６の態様に記載の透かし取出装置と、を具備した構成を採る。

【００２８】

このように第１の態様から第５の態様のいずれかに記載の透かし挿入装置と、第６の態様に記載の透かし取出装置と、を具備することにより、確実にプログラムの不正配布を防止できる。

【００２９】

本発明の第８の態様は、第７の態様にかかるプログラム不正配布防止システムにおいて、前記透かし挿入装置を前記配布先に設けたものである。

【００３０】

これにより、配布先に対して容易にプログラムを配布し、配布先において透かしを挿入するようにできる。このような形態は、単純にプログラムのみを配布することが好ましいシステムに効果的である。

【００３１】

本発明の第９の態様は、配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先に関わらず同じ透かし検証コードを前記プログラムに挿入するステップと、を具備したことを特徴とする透かし挿入方法である。

【００３２】

本発明の第１０の態様は、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報の挿入箇所の周辺もしくは前記プログラムの全体の仕様を維持したまま変換するステップと、を有することを特徴とする透かし挿入方法である。

【００３３】

これにより、プログラムの差分をとったときに、透かし以外の部分が差分として検出されるため、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【００３４】

本発明の第１１の態様は、コンピュータに、プログラムの配布先ごとに異なる透かし情報を前記プログラムに挿入するステップと、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記配布用プログラムを正しく動作させないもので

あり、前記配布先に関わらず同じ透かし検証コードを前記配布用プログラムに挿入するステップと、を行わせることを特徴とする透かし挿入プログラムである。

【００３５】

本発明の第１２の態様にかかる透かし挿入装置は、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入する透かし挿入手段と、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を維持したまま変換する変換手段と、を具備した構成を採る。

【００３６】

これにより、プログラムの差分をとったときに、透かし情報以外の部分が差分として検出されるため、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【００３７】

本発明の第１３の態様は、第１２の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えない実行コードの組を挿入する。

【００３８】

このように、仕様に影響を与えない実行コードの組を挿入することにより、差分攻撃が行われた際に透かし情報以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【００３９】

本発明の第１４の態様は、第１２の態様または第１３の態様にかかる透かし挿入装置において、前記透かし情報の挿入箇所を示す識別情報を記憶する。

【００４０】

これにより、識別情報を用いて透かし情報の挿入個所を容易に特定でき、透かし情報を容易に検出できる。

【００４１】

本発明の第１５の態様は、第１４の態様にかかる透かし挿入装置において、前記識別情報は、メソッド名もしくは行番号である。

【００４２】

これにより、識別情報により透かし情報の挿入個所を確実に検出できる。

【００４３】

本発明の第１６の態様は、第１２の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分に、仕様に影響を与えないように難読化処理する。

【００４４】

これにより、差分攻撃により透かし情報以外の部分が検出される。これにより、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。

【００４５】

本発明の第１７にかかる透かし取出装置は、第１２の態様から第１６の態様のいずれかに記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記プログラムから前記透かし情報を取り出す透かし検出手段と、を具備し、取り出した前記透かし情報に基づき配布先を特定する。

【００４６】

これにより、プログラムの配布先を特定することができ、プログラムの不正流通を防止できる。

【００４７】

本発明の第１８にかかる透かし取出装置は、第１５の態様または第１６の態様に記載の透かし挿入装置が前記透かし情報を挿入したプログラムを入力するプログラム入力手段と、前記識別情報を取得し、前記識別情報から透かし挿入箇所を特定し、特定した前記透かし挿入個所のみから前記透かし情報を取り出す透かし検出手段と、を具備し、取り出した

前記透かし情報に基づき配布先を特定する。

【００４８】

これにより、識別情報を用いることにより、透かし情報を容易に取り出すことができ、この透かし情報からプログラムの配布先を特定することができ、プログラムの不正流通を防止できる。

【００４９】

本発明の第１９の態様は、コンピュータに、プログラムの配布先ごとに異なる透かし情報をプログラムに挿入するステップと、前記透かし情報を挿入する箇所以外の部分を前記プログラムの仕様を変更することなく変換するステップと、行わせることを特徴とするプログラムである。

【００５０】

本発明の第２０の態様は、第１２の態様にかかる透かし挿入装置において、前記変換手段は、前記透かし情報を挿入する箇所以外の部分であって、順序を入れ替えても仕様に影響を与えない部分の順序を変換する。

【００５１】

これにより、差分攻撃によりプログラムの仕様に影響する部分が検出される。この結果、透かしの改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【００５２】

本発明の第２１の態様は、第２０の態様にかかる透かし挿入装置において、前記仕様に影響を与えない部分を変換した履歴情報を保持し、前記履歴情報を用いて、前記仕様に影響を与えない部分の変換を配布先ごとに異なるようにする。

【００５３】

これにより、仕様に影響を与えない部分の順序を、配布先ごとに確実に、かつ簡単に異なるようにできる。

【００５４】

（実施の形態１）

本発明の実施の形態１にかかる透かし挿入装置および透かし取出装置を具備したプログラム不正配布防止システムについて添付図面を用いて説明する。

【００５５】

図１は、実施の形態１にかかる透かし挿入による不正配布防止システムの構成図である。

【００５６】

まず、配布元１０は、プログラム配布の際に、透かし挿入装置２０により、配布先４０ａ、４０ｂごとに、異なる透かしを挿入して配布する（配布先の２次配布を認めないものとする）。

【００５７】

このように透かしを埋め込んで配布することで、不正な２次配布などプログラムが流出した際に、配布元１０は、透かし取出装置３０を用いて、流出先５０に流出したプログラムから透かしを取り出して配布先を確認し、流出元（配布先）４０ａ、４０ｂを特定することができる。

【００５８】

また、配布先４０ａ、４０ｂは、透かしによる流出元の特定を恐れて、不正な２次配布を控えることになる。

【００５９】

このようにして、不正配布防止システムは、透かしによる不正配布を抑止する。

【００６０】

次に、実施の形態１にかかる透かし挿入装置２０について図２を用いて説明する。図２は、実施の形態１の透かし挿入装置の構成図である。

【００６１】

透かし挿入装置２０には、プログラム入力部２０１が設けられている。プログラム入力

部２０１は、透かしを入力するプログラムコードを入力する手段である。プログラム入力部２０１は、プログラムコードを透かし挿入部２０２に出力する。

【００６２】

透かし挿入部２０２は、ＩＤ情報生成部２０５により生成されるＩＤ情報からプログラムに実際に埋め込む透かしを生成し、プログラム入力部２０１から出力されたプログラムコードに対し、透かしを入力する手段である。また、透かし挿入部２０２は、プログラム入力部２０１が出力したプログラムコードがソースコードであれば、ソースコードをコンパイルし、透かしの入力箇所をアセンブラコードの行番号として透かし用情報記憶部２０６に渡す。

【００６３】

プログラム出力部２０３は、透かし挿入部２０２が透かしを入力したプログラムコードを出力する手段である。

【００６４】

透かし用データ入力部２０４は、透かし用データを入力する。入力する透かし用データは、配布先を一意に特定する情報であり、配布先の住所、電話番号、会社名、氏名、電子メールアドレスなどである。また、透かし用データに、配布元の情報を入力してもよい。

【００６５】

ＩＤ情報生成部２０５は、透かし用データ入力部２０４により入力された透かし用データから一意に決定できるＩＤ情報を生成する。ＩＤ情報は、入力したデータそのものであってもよいし、それを暗号化したデータであってもよい。また、ＩＤ情報は、透かし用データを保存するデータベース上において透かし用データを一意に特定するためのＩＤであってもよい。

【００６６】

なお、本発明の実施の形態においては、ＩＤ情報に基づいて透かし情報を生成する形態となっているが、必ずしもＩＤ情報に基づいて透かし情報を生成する必要はなく、透かし情報から一意に配布先を特定可能となっていれば良い。例えば、ソフトウェアに１～Ｎシーケンス番号を透かし情報として挿入し、配布先Ａにシーケンス番号ｉのソフトを配布、配布先Ｂにシーケンス番号ｊのソフトを配布といったように透かし情報と配布先を一意に特定可能としてもよい。

【００６７】

透かし用情報記憶部２０６は、透かし挿入部２０２が挿入した透かしの挿入箇所を記憶する手段である。具体的には、透かしを挿入したコードのアセンブラコード行番号を記憶する。

【００６８】

次に、実施の形態１にかかる透かし取出装置３０について図３を用いて説明する。図３は、実施の形態１における、透かし取出装置３０の構成図である。

【００６９】

プログラム入力部３０１は、透かしを挿入したプログラムを入力する手段である。

【００７０】

透かし検出部３０２は、プログラム入力部３０１から出力されたプログラムを逆アセンブリングし、透かし用情報記憶部３０５より得られる透かし挿入箇所（アセンブラコード行番号）から入力された透かしを取り出す。そして、透かし検出部３０２は、取り出した透かしからＩＤ情報を生成し、ＩＤ情報記憶部３０４に渡す。

【００７１】

ＩＤ情報記憶部３０４は、透かし検出部３０２より得られるＩＤ情報から、配布先の情報を生成する手段である。ＩＤ情報記憶部３０４は、ＩＤ情報がデータベースのデータのＩＤである場合には、ＩＤからデータを取り出すことで、配布先の情報を取得する。また、ＩＤ情報記憶部３０４は、ＩＤ情報が配布先の情報の暗号化データである場合には、復号して配布先の情報を取得する。

【００７２】

透かし用情報記憶部３０５は、配布したプログラムの透かし挿入箇所を記憶している手段である。透かし挿入箇所の情報は、透かし挿入装置２０の透かし用情報記憶部２０６より得る。

【００７３】

出力部３０３は、取得された配布先の情報を出力する手段である。

【００７４】

次に、実施の形態１にかかる透かし挿入部２０２の透かし挿入動作について図４を用いて説明する。図４は、実施の形態１にかかる透かし挿入部２０２の動作を表すフローチャートである。

【００７５】

まず、透かし挿入部２０２は、配布先４０の情報から生成されるＩＤ情報Ｉから、実際にプログラムに挿入する透かし情報Ｘ１、Ｘ２を生成関数Ｆ（１）により生成する（ステップ４０１）。

【００７６】

続いて、透かし挿入部２０２は、透かし情報Ｘ１、Ｘ２を入力としたとき、定数Ｃ１を出力する関数Ｆ２１および定数Ｃ２を出力する関数Ｆ２２を構成する（ステップ４０２）。

【００７７】

続いて、透かし挿入部２０２は、透かし情報Ｘ１、Ｘ２を変数ｖａｌ１，ｖａｌ２に代入する式をプログラムコード中に埋め込む（ステップ４０３）。

【００７８】

続いて、透かし挿入部２０２は、プログラムコード中に、Ｆ２１（ｖａｌ１，ｖａｌ２）を変数ｖａｌ３に、Ｆ２２（ｖａｌ１，ｖａｌ２）を変数ｖａｌ４に代入する式を埋め込む（ステップ４０４）。

【００７９】

続いて、透かし挿入部２０２は、透かし検証コードである、変数ｖａｌ３と定数Ｃ１が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、変数ｖａｌ４と定数Ｃ２が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、をプログラムコード中に埋め込む（ステップ４０５）。

【００８０】

そして、透かし挿入部２０２は、ステップ４０３からステップ４０５において透かし情報および透かし検証コードを挿入した箇所を透かし用情報記憶部２０６に記憶する（ステップ４０６）。

【００８１】

このようにして、透かし挿入部２０２は、プログラムに透かしである透かし情報および透かし検証コードを挿入する。

【００８２】

なお、透かし挿入部２０２は、ステップ４０３からステップ４０５において、挿入した式および条件分岐（透かし検証コード）を、プログラムの実行順に入力する。ただし、Ｆ１は、Ｘ１、Ｘ２よりＩを一意に生成するＦ１の逆関数を持つことを条件とし、Ｆ２１、Ｆ２２は、Ｆ２１（Ｘ１，Ｘ２）＝＝Ｃ１かつＦ２２（Ｘ１，Ｘ２）＝＝Ｃ２をＸ１、Ｘ２以外の時には満たさないことを条件とする（“＝＝”は値が等しいことをあらわす。）。

【００８３】

たとえば、ＩＤ情報Ｉ＝１２３４５６７８、Ｆ１は８桁の値を４桁目から２つの値に分割する関数、Ｆ２１（ｘ，ｙ）、Ｆ２２（ｘ，ｙ）はａｘ＋ｂｙという２変数一次関数、Ｃ１＝２３４５、Ｃ２＝５６７８の場合を考える。

【００８４】

この場合、まず、Ｆ１から、透かし情報Ｘ１＝１２３４、Ｘ２＝５６７８が生成される。また、Ｆ２１、Ｆ２２は、ａ１×１２３４＋ｂ１×５６７８＝２３４５、ａ２×１２３

４＋ｂ２×５６７８＝５６７８を満たすａ１，ａ２，ｂ１，ｂ２を求めることにより構成する。たとえば、ａ１＝１，ａ２＝０．１９５６６７，ａ２＝３．７００９７２，ｂ２＝０．１９５６６７は条件を満たす。

【００８５】

次に、実施の形態１を適用した場合に生成されるプログラムの例を、図５に示す。

【００８６】

５００ａは、プログラム入力部２０１が入力した基本となる基本プログラムである。プログラム５００ｂ、５００ｃは、基本プログラム５００ａに透かしである透かし情報および透かし検証コードを入力した透かし挿入プログラムである。

【００８７】

まず、透かし挿入部２０２は、ステップ４０３において、透かし挿入プログラム５００ｂ、５００ｃに、それぞれ別のＩＤ情報Ｉａ（１２３４５６７８）、Ｉｂ（１１１１２２２２）より生成された透かし情報Ｘ１ａ（１２３４）、Ｘ１ｂ（５６７８）とＸ２ａ（１１１１）、Ｘ２ｂ（２２２２）を入力する（図中５０１に示す部分）。

【００８８】

次に、透かし挿入部２０２は、ステップ４０４において、透かし挿入プログラム５００ｂ、５００ｃに、それぞれ別のＦ２１、Ｆ２２を透かし挿入プログラム５００ｂ、５００ｃに挿入する（図中５０２に示す部分）。

【００８９】

そして、透かし挿入部２０２は、ステップ４０５において、透かし検証コードとして、変数ｖａｌ３と定数Ｃ１（２３４５）が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐（ａｓｓｅｒｔ（０））と、変数ｖａｌ４と定数Ｃ２（５６７８）が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐（ａｓｓｅｒｔ（０））と、をプログラムコード中に埋め込む（図中５０３に示す部分）。

【００９０】

ここで、着目すべき点は、２つのプログラム５００ｂと５００ｃの差分をとると、透かし情報である５０１、５０２の部分は検出されるが、透かし検証コードである条件分岐５０３は検出されないことである。これにより、プログラム５００ｂ、５００ｃに差分攻撃をすることにより、透かし入力箇所を検出し、検出部分の改ざん、削除を行ったとしても、透かし検証コードである条件分岐５０３の部分の改ざん、削除は行えない。よって、透かし検証コード５０３の部分が条件と合わなくなり、プログラムが動作しなくなる。

【００９１】

このように、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを削除した正常に動作するプログラムを入手できなくさせることが可能となる。

【００９２】

なお、わかりやすさのため、図５ではソースコードを用いて説明しているが、バイナリコードの場合でも同じことが言える。また、図５では、条件分岐５０３は、条件文が真であった場合にプログラムを停止するよう処理しているが、そうでなく、プログラムが異常な動作をするように（たとえば、ａ＋＋とするなど）プログラム中の変数値を変更するように処理することもできる。

【００９３】

さらに、実施の形態１では、ＩＤ情報から、２つの透かし情報を生成しているが、３つ以上の透かし情報を生成することとしてもよい。

【００９４】

次に、実施の形態１にかかる透かし検出部３０２の動作について図６を用いて説明する。図６は、実施の形態１の透かし検出部３０２の動作を表すフローチャートである。

【００９５】

まず、透かし検出部３０２は、プログラムの実行コードを逆アセンブルする（ステップ１００１）。

【００９６】
その後、透かし検出部３０２は、透かし用情報記憶部３０５を参照し、プログラムへの透かし挿入箇所を記憶した記憶情報（挿入箇所を示すアセンブラ行番号）を取得し、これに基づいて透かし情報Ｘ１、Ｘ２の入力箇所を特定する。そして、透かし検出部３０２は、透かし情報Ｘ１、Ｘ２をプログラムから取り出す（ステップ１００２）。
【００９７】
続いて、透かし検出部３０２は、透かし情報Ｘ１、Ｘ２を生成する際に使用した関数Ｆ１の逆関数を使用して、ＩＤ情報Ｉを生成する（ステップ１００３）。
【００９８】
このようにして、透かし検出部３０２は、ＩＤ情報Ｉを取得して、配布先４０の特定を行う。
【００９９】
なお、上記の方法では、実行コードの配布先もしくは流出先で、最適化、難読化といった処理によりコードの実行順番を入れ替えられた場合に、透かし情報の入力箇所のアセンブラ行番号が変化してしまい、透かし情報を得られない可能性がある。このような場合を考慮して、ステップ１００２の処理を、挿入箇所を示すアセンブラ行番号の周辺の行において、代入命令をさがし、代入命令のオペランド部を取り出すという処理に変更してもよい。
【０１００】
以上説明したように、実施の形態１によれば、透かし検証コード（図５の５０３の部分）は、配布先に関わらず同じなので、差分攻撃により、透かし検証コード（図５の５０３の部分）が差分として検出されないようにできる。これにより、差分攻撃により透かし検証コードの挿入位置を検出することができない。この結果、差分攻撃により検出された箇所だけを改変、削除するという単純な手法では、全ての透かしを改変、削除できなくなり、正常に動作する透かしのない（もしくは改変された）プログラムを生成することが不可能になる。よって、配布先は、透かしがなく、かつ正常に動作するプログラムを生成できないので、プログラムを不正に流通させることができなくなる。
【０１０１】
なお、透かし挿入装置２０および透かし取出装置３０の行う処理をプログラムにし、汎用のコンピュータに実行させる形態であってもよい。
【０１０２】
（実施の形態２）
本発明の実施の形態２は、不正にプログラムを配信しようとした者が、差分攻撃により透かし情報を検出し、検出した透かし情報に用いられている関数により生成される変数を使用している箇所（図５に示す５０３の部分）を検出し、検出した箇所を改変、削除することにより、実施の形態１の透かし検証コードを改変、削除しようとした場合に対応したものである。
【０１０３】
具体的には、透かし情報を用いたものであり、かつプログラムを正しく動作させるのに必要な透かし検証コードをプログラム中に挿入するようにしたものである。
【０１０４】
これにより、上述した手順により透かし情報を用いた透かし検証コードを検出し、改変、削除した場合には、プログラムを正常に動作させなくすることができる。
【０１０５】
以下、実施の形態２ついて詳細に説明する。実施の形態２における透かし挿入装置と、実施の形態１における透かし挿入装置２０との違いは、透かし挿入部２０２の動作である。
【０１０６】
次に、実施の形態２の透かし挿入部の動作について図７を用いて説明する。図７は、実施の形態２の透かし挿入部の動作のフローチャートである。

【０１０７】
ステップ６０１およびステップ６０２の動作は、実施の形態１で説明した図４のステップ４０１およびステップ４０２の動作と同様であるので説明を省略する。
【０１０８】
続いて、透かし挿入部は、透かし情報Ｘ１、Ｘ２より、Ｃ１＋Ｃ２＋Ｃ３＝０となるＣ３を生成する関数Ｆ３を生成する（ステップ６０３）。
【０１０９】
続いて、透かし挿入部は、透かし情報Ｘ１、Ｘ２を変数ｖａｌ１，ｖａｌ２に代入する式をプログラムコード中に埋め込む（ステップ６０４）。
【０１１０】
続いて、透かし挿入部は、プログラムコード中に、Ｆ２１（ｖａｌ１，ｖａｌ２）を変数ｖａｌ３に、Ｆ２２（ｖａｌ１，ｖａｌ２）を変数ｖａｌ４に代入する式を埋め込む（ステップ６０５）。
【０１１１】
続いて、透かし挿入部は、透かし検証コードとして、変数ｖａｌ３と定数Ｃ１が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、変数ｖａｌ４と定数Ｃ２が等しいかどうかを判定し、等しくない場合にはプログラムを停止する条件分岐と、をプログラムコード中に埋め込む（ステップ６０６）。
【０１１２】
続いて、透かし挿入部は、Ｆ３（ｖａｌ１，ｖａｌ２）を変数ｖａｌ５に代入する式を埋め込む（ステップ６０７）。
【０１１３】
次に、透かし挿入部は、透かし検証コードとして、ｖａｌ３＋ｖａｌ４＋ｖａｌ５をオリジナルコードの０を判定する判定文に加算するコードをプログラム中に挿入する（ステップ６０８）。
【０１１４】
そして、透かし挿入部２０２は、ステップ６０４からステップ６０８において透かし情報および透かし検証コードを挿入した箇所を透かし用情報記憶部２０６に記憶する（ステップ６０９）。
【０１１５】
このようにして、透かし挿入部は、プログラムに透かしを挿入する。
【０１１６】
ここで、着目すべき点は、ステップ６０８で挿入されるｖａｌ３＋ｖａｌ４＋ｖａｌ５には、差分検出で検出される変数、ｖａｌ３、ｖａｌ４、ｖａｌ５が含まれている点であり、ｖａｌ３＋ｖａｌ４＋ｖａｌ５がプログラムの動作に関わる判定文の０の部分に挿入されている点である。これにより、不正使用者が、差分攻撃により変数（ｖａｌ３、ｖａｌ４、ｖａｌ５）を検出し、検出した変数を用いている関数により生成される変数を使用している箇所を改変、削除しようとした場合に、プログラムの動作にかかわる判定文も改変、削除してしまうことになる。よって、プログラムが正常に動作しなくなり、不正使用もできなくなる。
【０１１７】
次に、実施の形態２にかかる透かし挿入部が生成したプログラムについて図８を用いて説明する。
【０１１８】
８００ａは、プログラム入力部２０１が入力した基本となる基本プログラムである。プログラム８００ｂは、基本プログラム８００ａに透かしを入力した透かし挿入プログラムである。
【０１１９】
プログラム８００ｂには、ステップ６０４において、７０１で示される部分に透かし情報が挿入され、ステップ６０５、ステップ６０７において、７０２で示される部分に透か

し検証用の計算式（コード）が挿入される。

【０１２０】

そして、プログラム８００ｂには、７０３で示される部分に、ステップ６０８の処理結果が挿入される。また、プログラム８００ｂには、ステップ６０６において、７０４で示される部分に透かし検証コードが挿入される。

【０１２１】

このようにプログラム８００ｂを生成することにより、不正に使用しようとした者が、差分攻撃によりプログラム８００ｂから透かし検証コード７０３を検出し、透かし検証コードを改変、削除すると、透かし検証コード７０３は、仕様（オリジナルの祖コードにおける、プログラムの入出力関係）に関連するコードであるので、このコードを削除するとプログラムが正しく動作しなくなる。

【０１２２】

透かしのうち、透かし検証コード７０３の判定文のみ変更しようとするためにはプログラムの仕様を理解して、透かし検証コード７０３が仕様に関連あるコードであることを知る必要がある。これは、プログラムの構造を理解した物が、時間をかけて行う必要があり、機械的な処理により透かしの削除はできない。

【０１２３】

なお、Ｃ１＋Ｃ２＋Ｆ３＝０でなくても良い。ただし、この場合は、Ｃ１＋Ｃ２＋Ｆ３により得られる値を用いた判定文にＣ１＋Ｃ２＋Ｆ３を挿入すればよい。例えば、Ｃ１＋Ｃ２＋Ｆ３＝１である場合は、１を判定する判定文の１をＣ１＋Ｃ２＋Ｆ３を置換する。

【０１２４】

以上説明したように、実施の形態２によれば、差分攻撃により検出した透かし情報（７０１、７０２）に用いられている関数により生成される変数を使用している箇所（図８に示す７０３の部分）を検出し、改変、削除した場合に、プログラムが正常に動作しないようになる。つまり、正常に動作する透かしのない（もしくは改変された）プログラムを生成することを不可能にすることができるので、プログラムの不正配布を防止できる。

【０１２５】

（実施の形態３）

本発明の実施の形態３は、透かし情報および透かし検証コードを入力した箇所周辺のコード、もしくはコード全体を難読化などの処理をすることにより改変するものである。これにより、差分攻撃により透かし以外のコードが検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【０１２６】

以下、実施の形態３について詳細に説明する。実施の形態３における透かし挿入装置と、実施の形態１における透かし挿入装置２０との違いは、透かし挿入部２０２の動作である。

【０１２７】

次に、実施の形態３の透かし挿入部２０２の動作について図９を用いて説明する。図９は、実施の形態３の透かし挿入部２０２の動作フローチャートである。

【０１２８】

まず、透かし挿入部２０２は、変数ｉに初期値１を代入する（ステップ８００）。次に、透かし挿入部は、ＩＤ情報をｎ個の情報に分割して、透かし情報Ｘ（１），Ｘ（２）…Ｘ（ｎ）を生成する（ステップ８０１）。

【０１２９】

続いて、透かし挿入部２０２は、プログラムソースコード中のループ部（ｗｈｉｌ，ｆｏｒ文）を検出し（ステップ８０２）、ループの内部に透かし情報Ｘ（ｉ）を挿入する（ステップ８０３）。

【０１３０】

その後、透かし挿入部２０２は、“ループを含むプログラムを難読化する方法の提案”，（門田ら，信学論Ｄ－Ｉ，　Ｖｏｌ．Ｊ８０－Ｄ－Ｉ，　Ｎｏ．７，　ｐｐ．６４４－

６５２，　Ｊｕｌｙ　１９９７）に記載される方法を適用することで、挿入箇所のループ部を難読化する（ステップ８０４）。この際、プログラムの難読化の仕方に複数のバリエーションがあるが、バリエーションをランダム（もしくは過去に配布したプログラムに施した難読化と重ならないよう）に選択する。

【０１３１】

そして、透かし挿入部２０２は、変数ｉが透かし情報の数ｎ以下か判断し（ステップ８０５）、変数ｉがｎ以下の場合は変数ｉをインクリメントし（ステップ８０６）、ステップ８０２の処理に以降する。一方、ステップ８０５において、変数ｉがｎ以下でないと判断した場合、つまり全ての透かし情報を入力した場合は、透かし挿入部２０２は、その後ソースコードをコンパイルし、透かし情報が入力されたアセンブラコードの行番号を記憶してプログラムを出力し、処理を終了する（ステップ８０７）。

【０１３２】

次に、実施の形態３にかかる透かし挿入部２０２が生成したプログラムについて図１０を用いて説明する。９００ａは、プログラム入力部２０１が入力した基本となる基本プログラムである。プログラム９００ｂ、９００ｃは、基本プログラム９００ａに透かし情報９０１を入力した透かし挿入プログラムである。

【０１３３】

プログラム９００ｂ、９００ｃには、難読化により、実装は異なるものの、仕様（プログラムの入出力の関係）は変化していない。プログラム９００ｂ、９００ｃの差分をとると、透かし以外の場所もプログラムコードが変化しているため、透かし以外の部分９０２ａ、９０２ｂも差分として検出される。

【０１３４】

したがって、プログラム９００ｂ、９００ｃの透かしを改変、削除するためには、プログラムを解析し、どの部分がプログラムの仕様に関係のない透かしとなっているかを探し出す必要がある。プログラムの仕様に関係ない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。

【０１３５】

以上説明したように、実施の形態３は、透かし挿入部が、プログラムの透かしを挿入する箇所以外の部分に、プログラムの仕様に影響を与えないように難読化処理をする改変手段としての動作もするので、差分攻撃により透かし以外のコードであるプログラムの仕様に関係する部分が検出される。これにより、差分攻撃に基づいて透かし挿入位置を特定することが困難になる。この結果、透かし改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【０１３６】

（実施の形態４）

本発明の実施の形態４は、配布先に透かし挿入装置を保持し、配布したプログラムに対して配布先で透かしを付与するものである。

【０１３７】

以下、実施の形態４にかかる不正配布防止システムの構成について図１１を用いて説明する。図１１は、実施の形態４にかかる透かし挿入による不正配布防止システムの構成図である。なお、すでに説明した部分と同一の部分には同一の符号を付与してある。

【０１３８】

本システムでは、まず、配布元１１００が、配布先１１１０、１１２０に対して、それぞれ配布先１１１０、１１２０を一意に決定するＩＤ情報１１０１，１１０２を配布する。

【０１３９】

これに対して、配布先１１１０、１１２０の透かし挿入装置２０ａ、２０ｂでは、ＩＤ情報１１０１、１１０２を記憶しておく。

【０１４０】

続いて、配布元１１００は、配布先１１１０、１１２０にプログラム１１０３を配布する。

【０１４１】

これに対して、配布先１１１０、１１２０では、配布されたプログラム１１０３に対して、透かし挿入装置２０ａ、２０ｂを利用して透かしを挿入したプログラム１１１１、１１２１を生成する。

【０１４２】

なお、透かし挿入装置２０ａ、２０ｂは、実施の形態１から実施の形態３のいずれかにかかるものであってもよい。

【０１４３】

その後、透かし挿入装置２０ａ、２０ｂは、透かしを挿入した箇所を記憶した記憶情報１１０４、１１０５を配布元１１００に送信し、配布元１１００では記憶情報１１０４、１１０５を保存する。

【０１４４】

配布元１１００は、配布先１１１０が、流出先１１３０に不正な２次配布を行った場合には、流出したプログラム１１１２を取得し、記憶情報１１０４、１１０５とともに透かし取出装置３０に入力する。そして、配布元１１００は、透かし取出装置３０において、配布先１１１０、１１２０を特定するＩＤ情報１１０７を入手する。そして、配布元１１００は、配布先１１１０、１１２０に配布したＩＤ情報１１０１、１１０２と、入手したＩＤ情報１１０７を比較し、不正にプログラムを流出した配布先１１１０、１１２０を特定する。

【０１４５】

以上説明したように、実施の形態４によれば、不特定多数の配布先に対して容易にプログラムを配布し、配布先において透かしを挿入するようにできる。このような形態は、単純にプログラムのみを配布することが好ましいシステム、たとえばデジタル放送を利用したプログラムの配布であるとか、ＩＰネットワークでマルチキャスト、ブロードキャストを利用したプログラム配布などに適用すると効果的である。

【０１４６】

（実施の形態５）

本発明の実施の形態５は、透かしを挿入するメソッドやその他のメソッドが実装されている箇所に、プログラムの仕様に影響を与えない偽装コードを追加することにより、プログラムを改変するものである。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。

【０１４７】

次に、実施の形態５にかかる透かし挿入装置１２００について図１２を用いて説明する。図１２は、実施の形態５の透かし挿入装置の構成図である。

【０１４８】

実施の形態５にかかる透かし挿入装置１２００のプログラム入力部２０１は、他の実施の形態における透かし挿入装置２０のプログラム入力部２０１と同じ動作をする。

【０１４９】

透かし挿入装置１２００には、プログラム入力部２０１により出力されたプログラムの実行に影響を与えない余分なダミーメソッドを入力するダミーメソッド入力部１２０３が設けられている。ダミーメソッド入力部１２０３は、入力したダミーメソッドをダミーメソッド挿入部１２０１に出力する。

【０１５０】

ダミーメソッド挿入部１２０１は、ダミーメソッド入力部１２０３で入力されたダミーメソッドを、透かしを埋め込むための領域として追加する手段である。ダミーメソッド挿入部１２０１は、ダミーメソッドの追加されたプログラムを偽装コード挿入部１２０２に出力する。

【０１５１】
偽装コード挿入部１２０２は、ダミーメソッド挿入部１２０１から出力されたプログラム全体のメソッド（ダミーメソッドを含む全てのメソッド）が実装されている箇所に、プログラムの実行に影響を与えず、プログラムの実行結果には必要としない偽装コードの組を挿入することにより、プログラムの仕様を変更することなく改変する改変手段である。挿入する偽装コードとしては、ＰＵＳＨとＰＯＰの組などが考えられる。
【０１５２】
透かし挿入部２０２、プログラム出力部２０３、透かし用データ入力部２０４、およびＩＤ情報生成部２０５は、他の実施の形態における透かし挿入装置２０の透かし挿入部２０２とプログラム出力部２０３、透かし用データ入力部２０４、およびＩＤ情報生成部２０５と、それぞれ同じ手段である。
【０１５３】
透かし用情報記憶部１２０４は、透かし挿入部２０２が挿入した透かしに対して、透かしとして用いた文字や数値、および記号とビット列との対応情報と、ビット列と命令コードとの対応情報を記憶する。また、透かし用情報記憶部１２０４は、透かしを挿入したダミーメソッドの識別情報として、メソッド名や行番号を保存する。さらに、透かし用情報記憶部１２０４は、透かし用データとして暗号化したデータを使用した場合には、データを復号するための鍵の情報をあわせて記憶する。
【０１５４】
これにより、識別情報を用いて、透かしの挿入個所を容易に特定でき、透かし情報を容易に検出できる。
【０１５５】
次に実施の形態５にかかる透かし取出装置３０について説明する。実施の形態５における透かし取出装置３０と、他の実施の形態における透かし取出装置３０との違いは、透かし検出部３０２、透かし用情報記憶部３０５の動作である。
【０１５６】
透かし検出部３０２は、プログラム入力部３０１から出力されたプログラム中で、透かし用情報記憶部３０５から得られる、透かしを挿入したメソッドの識別情報を獲得し、識別情報の表すメソッドを検査する。
【０１５７】
次に、透かし検出部３０２は、同じ透かし用情報記憶部３０５から得られる、透かしとして用いた文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応を利用して、命令コードからビット列、ビット列から文字や数値、および記号へ変換することにより、プログラムに挿入されている透かし情報を取り出す。
【０１５８】
透かし検出部３０２は、取り出した透かしからＩＤ情報を生成し、ＩＤ情報記憶部３０４に出力する。
【０１５９】
透かし用情報記憶部３０５は、透かしの挿入されているメソッドの識別情報を保持している手段である。また、透かし用情報記憶部３０５は、配布したプログラムの透かしとして用いた文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応も記憶している。さらに、透かし用情報記憶部３０５は、挿入された透かしのデータが暗号であった場合に、暗号を復号するための鍵も保持している。透かし用情報記憶部３０５は、文字や数値、および記号とビット列との対応と、ビット列と命令コードとの対応、透かしの挿入されているメソッドの識別情報、暗号データを復号するための鍵は、透かし挿入装置１２００の透かし用情報記憶部１２０４より得る。
【０１６０】
次に、実施の形態５の偽装コード挿入部１２０２と透かし挿入部２０２の動作について図１３を用いて説明する。図１３は、実施の形態５の偽装コード挿入部１２０２と透かし挿入部２０２の動作フローチャートである。

【０１６１】
まず、偽装コード挿入部１２０２は、変数ｉに初期値１を代入する（ステップ１３００）。次に、透かし挿入部２０２は、文字や数値、および記号とビット列の対応を用いて、ＩＤ情報より透かし情報Ｓを生成する（ステップ１３０１）。
【０１６２】
続いて、偽装コード挿入部１２０２は、プログラム中で、メソッドが実装されている箇所であるメソッド部を検出し（ステップ１３０２）、変数ｉがプログラム中の総メソッド数以下であるかの判断を行い（ステップ１３０３）、変数ｉが総メソッド数以下である場合には、プログラムの仕様に影響を与えない、本来は必要ではない偽装コードを挿入する（ステップ１３０４）。
【０１６３】
このとき挿入する偽装コードは、複数のバリエーションがあるが、バリエーションをランダム、もしくは過去に配布したプログラムに挿入した偽装コードと重複しないように選択する。つまり、差分攻撃により、偽装コードが抽出されるように偽装コードを挿入する。
【０１６４】
次に、透かし挿入部２０２は、検出されたメソッド部がダミーメソッドであるかを判断し（ステップ１３０５）、ダミーメソッドである場合には、“プログラムに電子透かしを挿入する一手法”、（門田ら、１９９８年暗号と情報セキュリティシンポジウム、ＳＣＩＳ’９８-９．２．Ａ，　Ｊａｎ．　１９９８）を適用することで、透かし情報Ｓを挿入する（ステップ１３０６）。
【０１６５】
また、このとき、透かし挿入部２０２は、ダミーメソッドの識別情報を保存する（ステップ１３０７）。
【０１６６】
そして、透かし挿入部２０２は、変数ｉをインクリメントし（ステップ１３０８）、ステップ１３０２の処理に移行する。
【０１６７】
一方、ステップ１３０３において、変数ｉがプログラムコードの総メソッド数以下でないと判断した場合、つまり全てのメソッドに偽装コードを挿入し、そのうちのダミーメソッドに透かし情報を挿入した場合は、透かし挿入部２０２は、透かし情報が埋め込まれたプログラムを出力する（ステップ１３０９）。
【０１６８】
実施の形態５にかかる透かし挿入部２０２が生成したプログラムは、偽装コードの挿入により、実装は異なるものの、仕様（プログラムの入出力の関係）は変化していない。また、それぞれのプログラムで異なる偽装コードが挿入されているため、透かしの挿入メソッドを特定するためにプログラム間の差分をとると、透かしが挿入されているメソッド以外のメソッドも差分として検出される。
【０１６９】
したがって、プログラムの透かしを改変、削除するためには、プログラムを解析し、どのメソッドがプログラムの仕様に関係のない透かし挿入用のダミーメソッドとなっているかを探し出す必要がある。プログラムの仕様に関係ない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。
【０１７０】
次に、実施の形態５を適用した場合に生成されるプログラムの例を、図１４に示す。
【０１７１】
図中１６００ａで示されるプログラムは、基本となるソースプログラムである。このプログラム１６００ａをコンパイルしたものが、プログラム入力部２０１より透かし挿入装置１２００に入力されるプログラムであるが、ここでは、説明の管理化のために、コンパ

イルしたプログラム１６００ａを逆アセンブルしたプログラム１６００ｂを用いて説明を行う。

【０１７２】

また、プログラム１６００ｃとプログラム１６００ｄは、それぞれ異なる透かしと偽装コードを挿入したプログラムである。またそれぞれのプログラム１６００ａ～１６００ｄにおいて、Ａ２のメソッドがダミーメソッドを表し、命令ニーモニックの前にある数字はメソッドごとの行番号を示している。

【０１７３】

まず、透かし挿入部２０２は、ステップ１３０１において、透かし挿入プログラム１６００ｃ、１６００ｄ用に、それぞれ別のＩＤ情報Ｉ１（（Ｃ）０１）、Ｉ２（（Ｃ）０２）より、１文字６ビットで透かし情報Ｓ１（１００１１１　００１１０１　１０１０００　０００００　０００００１）、Ｓ２（１００１１１　００１１０１　１０１０００　０００００　０００００１０）を生成する。

【０１７４】

次に、偽装コード挿入部１２０２は、ステップ１３０２において、透かし挿入プログラム１６００ｂで、メソッド部を検出し、ステップ１３０４においてダミーメソッドではないＡ１にそれぞれ異なる偽装コードを挿入する（図１４中の１６０１で示される部分）。

【０１７５】

さらに、透かし挿入部２０２は、メソッドがダミーメソッドＡ２の場合には、ステップ１３０６において、透かし挿入プログラム１６００ｂに対して、透かし情報Ｓ１とＳ２から埋め込み対象の命令に割り当てられたビット数分だけ透かし情報として埋め込む。

【０１７６】

この例では、プログラム１６００ｂのメソッドＡ２内のｉｃｏｎｓｔ＿０が埋め込み対象命令で、２ビットの情報量が割り当てられており、Ｓ１とＳ２から２ビットが取り出されて埋め込みが行われる（図１４の１６０２で示される部分）。

【０１７７】

このとき、透かし挿入部２０２は、各文字の下位ビットから取り出しを行い、一文字分全て取り出し終わった場合には、次の文字の下位ビットから取り出しを行う。

【０１７８】

また、偽装コード挿入部１２０２は、メソッドＡ２に対しても、メソッドＡ１と同じように偽装コードの挿入も行う（図１４の１６０３で示される部分）。

【０１７９】

もし、プログラム１６００ｃと１６００ｄの配布先が結託し、透かし情報の挿入場所を特定するために、プログラム間の差分をとったとしても、透かし情報ではない１６０１で示される部分や１６０３で示される部分が、透かし情報の１６０２とともに検出されてしまうため、差分攻撃に基づいて透かし情報の挿入位置を特定することが困難となる。

【０１８０】

したがって、透かし情報の機械的な改変、削除を確実に防ぐことが可能で、プログラムの不正流通を防止することができる。

【０１８１】

次に、実施の形態５にかかる透かし検出部３０２の動作について図１５を用いて説明する。図１５は、実施の形態５の透かし検出部３０２の動作を表すフローチャートである。

【０１８２】

まず、透かし検出部３０２は、透かし用情報記憶部３０５よりダミーメソッドの識別情報を獲得する（ステップ１５００）。

【０１８３】

続いて、透かし検出部３０２は、獲得した識別情報を用いて、プログラム中でダミーメソッドが実装されているダミーメソッド部とメソッド部を検出し（ステップ１５０１）、透かし用情報記憶部３０５に記憶しておいたビット列と命令コードの対応を用いて、ダミーメソッド部から透かし情報Ｓを取り出す（ステップ１５０２）。

【０１８４】
さらに、透かし検出部３０２は、ＩＤ情報記憶部３０４で記憶された情報と、取り出した透かし情報Ｓから、プログラムの配布先を一意に特定するＩＤ情報を生成し（ステップ１５０３）、ＩＤ情報を出力（ステップ１５０４）して終了する。
【０１８５】
このように、透かし検出部３０２は、識別情報を用いることにより、ダミーメソッド部とメソッド部を容易に検出し、ダミーメソッド部から透かし情報Ｓを取り出すことにより、プログラムの配布先を特定できる。この結果、プログラムの不正流通を防止できる。
【０１８６】
以上説明したように、実施の形態５によれば、プログラム中に透かし情報だけでなく、仕様に影響を与えない実行コードの組である偽装コードを挿入することができる。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されてしまうので、差分攻撃に基づく透かし改変、削除を確実に防ぐことができる。
【０１８７】
（実施の形態６）
本発明の実施の形態６は、透かし情報の挿入箇所以外の一部もしくはプログラム全体のコードの順序を入れ替えることにより、プログラムを改変するものである。これにより、差分攻撃が行われた際に透かし以外のコードが異なった箇所として検出されるので、差分攻撃に基づく透かしの改変、削除を確実に防ぐことができる。
【０１８８】
以下、実施の形態６について詳細に説明する。実施の形態６における透かし挿入装置と、実施の形態１における透かし挿入装置２０との違いは、透かし挿入部２０２の動作である。
【０１８９】
次に、実施の形態６の透かし挿入部の動作について図１６を用いて説明する。図１６は、実施の形態６の透かし挿入部の動作フローチャートである。
【０１９０】
まず、透かし挿入部２０２は、変数ｉに初期値１を代入する（ステップ１６０１）。次に、透かし挿入部２０２は、コード（プログラム）中に埋め込むための（配布先ごとに異なる）透かし情報ＳをＩＤ情報より生成する（ステップ１６０２）。
【０１９１】
続いて、透かし挿入部２０２は、コード全体で、順序を入れ替えても仕様に影響を与えない、つまり仕様を維持できるコード部分を抽出する（ステップ１６０３）。ここでコード部分とは、複数のコードからなるプログラムの一部をいう。
【０１９２】
そして、透かし挿入部２０２は、変数ｉが、順序を入れ替えても仕様を維持できるコード部分の数Ｎ以下であるか判断し（ステップ１６０４）、変数ｉがＮ以下であった場合には、そのコード部分に含まれるコードの順序を入れ替え（ステップ１６０５）、ｉをインクリメントして（ステップ１６０６）、ステップ１６０４に移行する。
【０１９３】
変数ｉがＮ以下でない場合には、透かし挿入部２０２は、透かし情報Ｓをコード中に挿入し（ステップ１６０７）、ソースコードをコンパイルし、透かし情報Ｓが挿入されたアセンブラコードの行番号を記憶してプログラムを出力し、処理を終了する（ステップ１６０８）。
【０１９４】
このように、透かし挿入部２０２は、順序を入れ替えても仕様を維持できるコード部分の順序を入れ替えることにより、透かし情報を挿入する個所以外の部分をプログラムの仕様を維持したまま変換する。
【０１９５】
次に、実施の形態６にかかる透かし挿入部が生成したプログラムについて図１７を用い

て説明する。プログラム１７００ａは、プログラム入力部２０１により入力されるオリジナルのプログラムである。また、プログラム１７００ｂ、１７００ｃは、オリジナルプログラム１７００ａに配布先ごとに異なる透かし１７０２ｂ、１７０２ｃを挿入したプログラムである。

【０１９６】

また、プログラム１７００ｂ、プログラム１７００ｃは、コード部分１７０１ｂ、１７０１ｃ、および１７０３ｂ、１７０３ｃを含むプログラムである。コード部分１７０１ｂ、１７０１ｃ、および１７０３ｂ、１７０３ｃは、オリジナルのプログラム１７００ａに含まれる透かしを挿入するコード部分でなく、かつコードを並び替えても仕様を維持できるコード部分１７０１ａ、１７０３ａのコードの並びを変えたものである。

【０１９７】

このように、プログラム１７００ｂ及びプログラム１７００ｃは、プログラム１７００ａに対してそれぞれ異なる命令順序に変換されているが、全体の仕様は変化していない。つまり、プログラム１７００ｂ及びプログラム１７００ｃは、プログラム１７００ａは、仕様を維持したまま変換されている。ここでプログラム１７００ｂ、１７００ｃの差分をとると、透かし以外の場所もプログラムコードが変化しているため、透かし以外の部分であるコード部分１７０１ｂ、１７０１ｃ、１７０３ｂ、１７０３ｃの部分も差分として検出される。

【０１９８】

したがって、プログラム１７００ｂ、１７００ｃの透かしを改変、削除するためには、プログラムを解析し、どの部分がプログラムの仕様に影響のない透かしとなっているかを探し出す必要がある。プログラムの仕様に影響のない部分であるかどうかを判定するためには、プログラムの仕様を理解する必要があるため、この方法により埋め込まれた透かしを機械的に削除することは困難となる。

【０１９９】

以上説明したように、実施の形態６は、透かし挿入部２０２が、プログラムの透かしを挿入する箇所以外のプログラム部分のうち、命令順序を入れ替えても仕様を維持できるプログラム部分を検出し、プログラムの仕様に影響を与えないように、つまり仕様を維持したまま、命令順序を入れ替えても良いプログラム部分の順序変換する変換手段としての動作をする。これにより、差分攻撃により透かし以外のコードであるプログラムの仕様に影響するプログラム部分が検出される。この結果、透かしの改変、削除を確実に防ぐことができ、プログラムの不正流通を防止できる。

【０２００】

また、実施の形態６によれば、命令順序を入れ替えても良いプログラム部分の順序変換を、プログラム部分中の命令文の順列を求め、配布先ごとに異なるように選択された順列に従い変換する。これにより、配布先ごとに命令順序を入れ替えても良いプログラム部分の命令順序が異なるようになる。よって、命令順序を入れ替えても良いプログラム部分の特定が困難になり、確実に透かしの改変、削除を確実に防ぐことができる。

【０２０１】

また、命令文の順列に従う変換以外の方法として、命令順序を入れ替えてもよいプログラム部分の順序をランダムに変換して、配布先ごとに異なるようにしてもよい。

【０２０２】

なお、順序を入れ替えてもよいコード部分に含まれるコードの順序変換の履歴情報を保持しておき、この履歴情報を用いて、順序を入れ替えてもよいコード部分の変換を配布先ごとに異なるようにしても良い。

【０２０３】

これにより、順序を入れ替えてもよいコード部分に含まれるコードの順序変換を、配布先ごとに確実に、かつ簡単に異なるようにできる。

【産業上の利用可能性】

【０２０４】

以上説明したように、本発明によれば、透かしの挿入箇所を特定することが困難なように透かしを挿入できるので、ネットワークを利用したコンピュータプログラムの流通等に広範囲に適応できる。

【図面の簡単な説明】

【０２０５】

【図１】本発明の実施の形態１にかかる透かし挿入による不正配布防止システムの構成図

【図２】実施の形態１の透かし挿入装置の構成図

【図３】実施の形態１における、透かし取出装置の構成図

【図４】実施の形態１にかかる透かし挿入部の動作を表すフローチャート

【図５】実施の形態１を適用した場合に生成されるプログラムを示す図

【図６】実施の形態１の透かし検出部の動作を表すフローチャート

【図７】本発明の実施の形態２にかかる透かし挿入部の動作のフローチャート

【図８】実施の形態２にかかる透かし挿入部が生成したプログラムを示す図

【図９】本発明の実施の形態３にかかる透かし挿入部の動作フローチャート

【図１０】実施の形態３にかかる透かし挿入部が生成したプログラムを示す図

【図１１】本発明の実施の形態４にかかる透かし挿入による不正配布防止システムの構成図

【図１２】本発明の実施の形態５における透かし挿入装置の構成図

【図１３】実施の形態５における偽装コード挿入部および透かし挿入部の動作のフローチャート

【図１４】実施の形態５における透かし挿入部が生成したプログラムの例

【図１５】実施の形態５における透かし検出部の動作のフローチャート

【図１６】本発明の実施の形態６の透かし挿入部の動作フローチャート

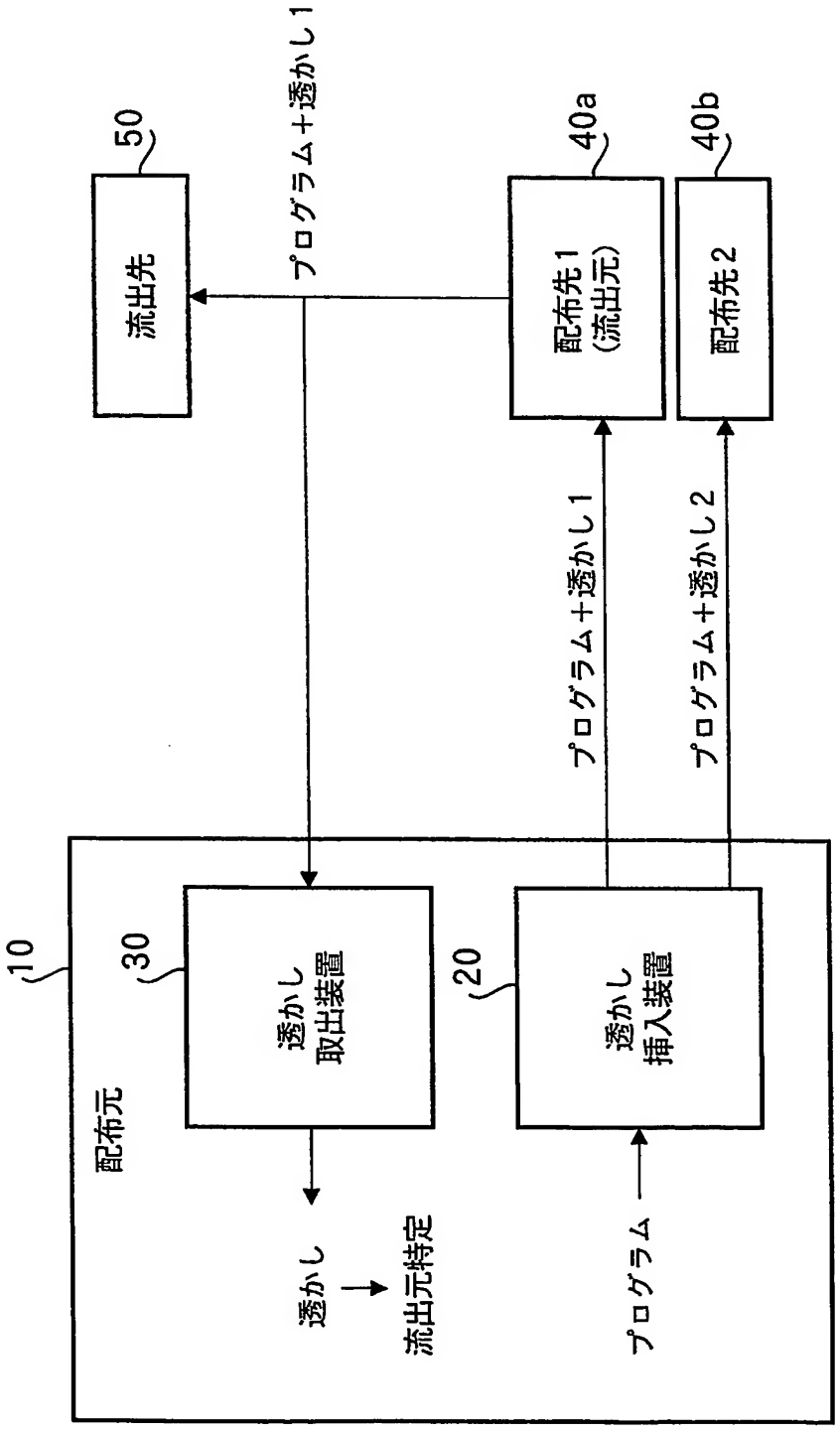
【図１７】実施の形態６にかかる透かし挿入部が生成したプログラムを示す図

【符号の説明】

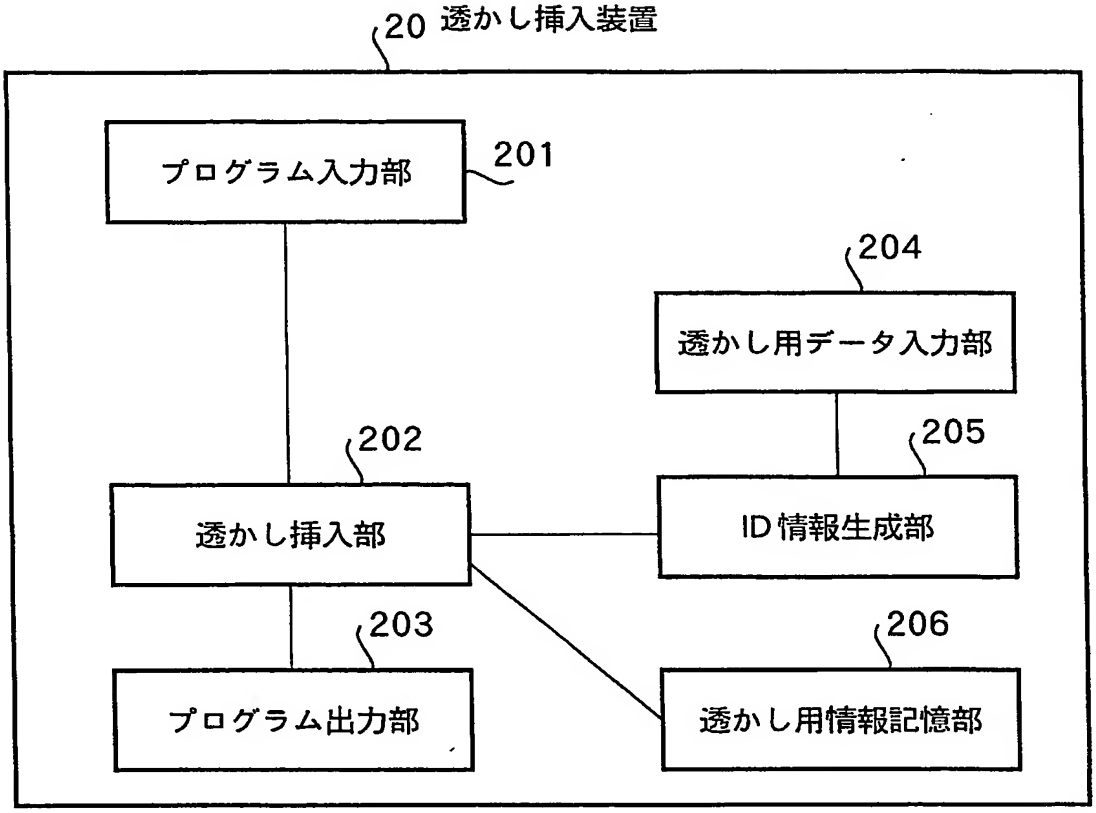
【０２０６】

１０、１１００　配布元
２０、１２００　透かし挿入装置
３０　透かし取出装置
４０ａ、４０ｂ、１１１０、１１２０　配布先
５０、１１３０　流出先
２０１、３０１　プログラム入力部
２０２　透かし挿入部
２０３　プログラム出力部
２０４　透かし用データ入力部
２０５　ＩＤ情報生成部
２０６、３０５、１２０４　透かし用情報記憶部
３０２　透かし検出部
３０３　出力部
３０４　ＩＤ情報記憶部
１２０１　ダミーメソッド挿入部
１２０２　偽装コード挿入部
１２０３　ダミーメソッド入力部

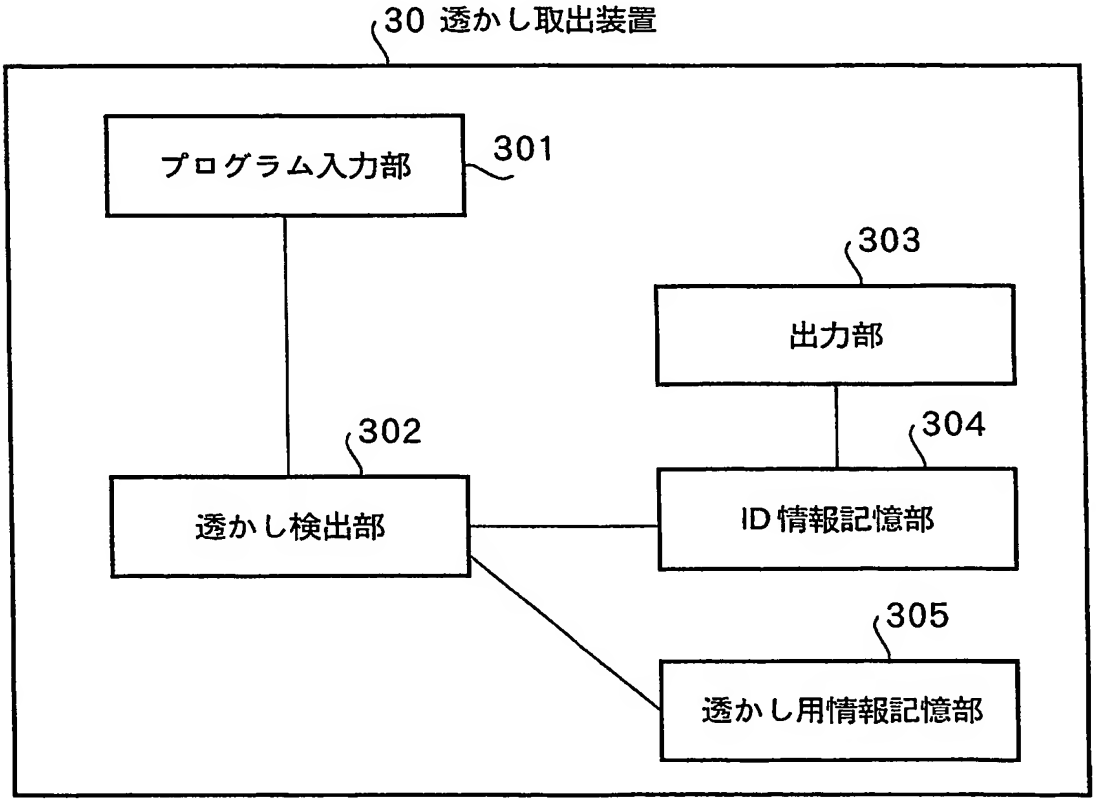
【書類名】図面
【図１】


50
流出先
プログラム＋透かし１
40a
配布先１
(流出元)
40b
配布先２
プログラム＋透かし１
プログラム＋透かし２
10
配布元
30
透かし
取出装置
20
透かし
挿入装置
透かし
流出元特定
プログラム

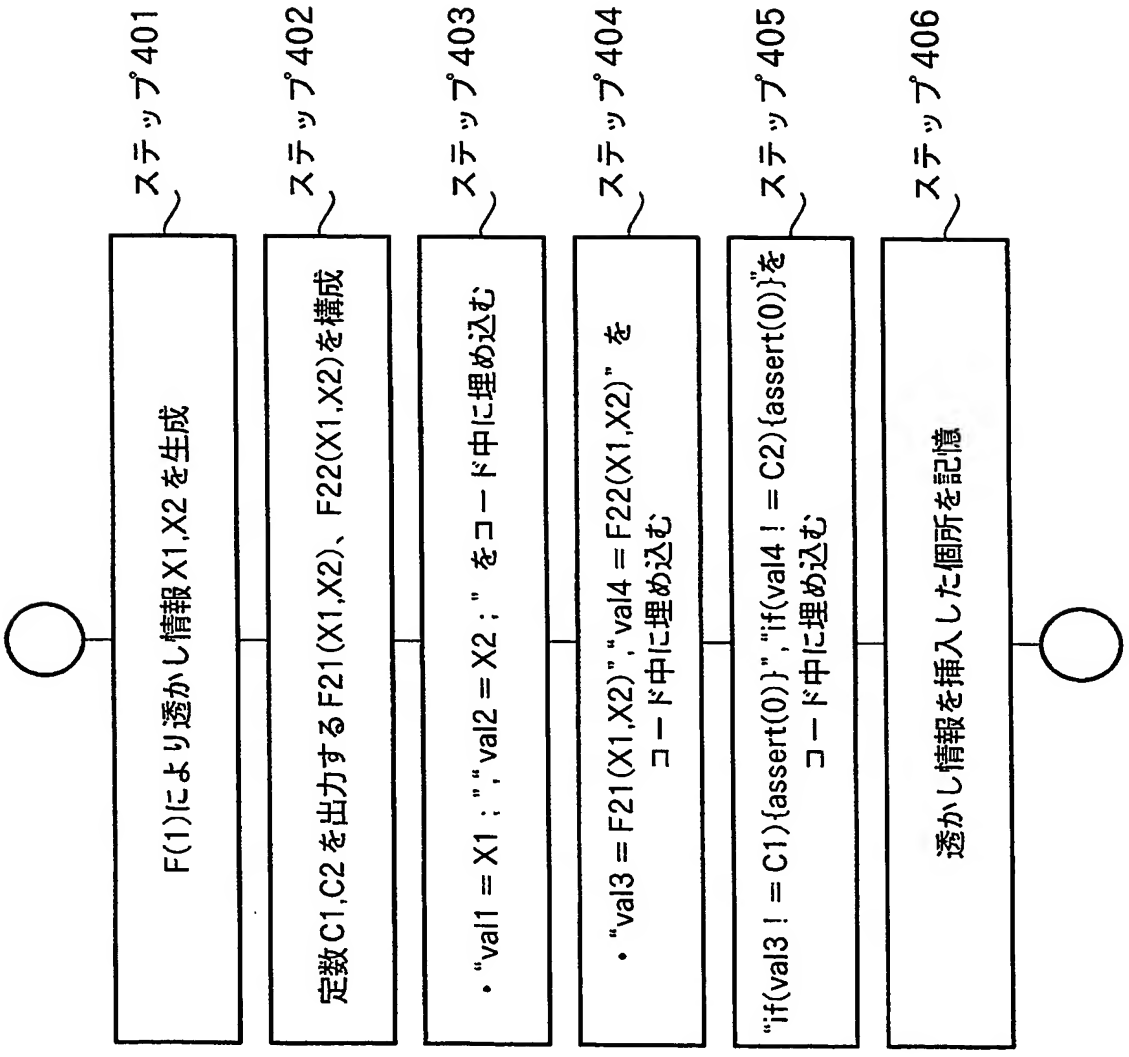
【図２】


20 透かし挿入装置
プログラム入力部
201
204
透かし用データ入力部
202
205
透かし挿入部
ID情報生成部
203
206
プログラム出力部
透かし用情報記憶部

【図３】


30 透かし取出装置
プログラム入力部
301
303
出力部
302
304
透かし検出部
ID情報記憶部
305
透かし用情報記憶部

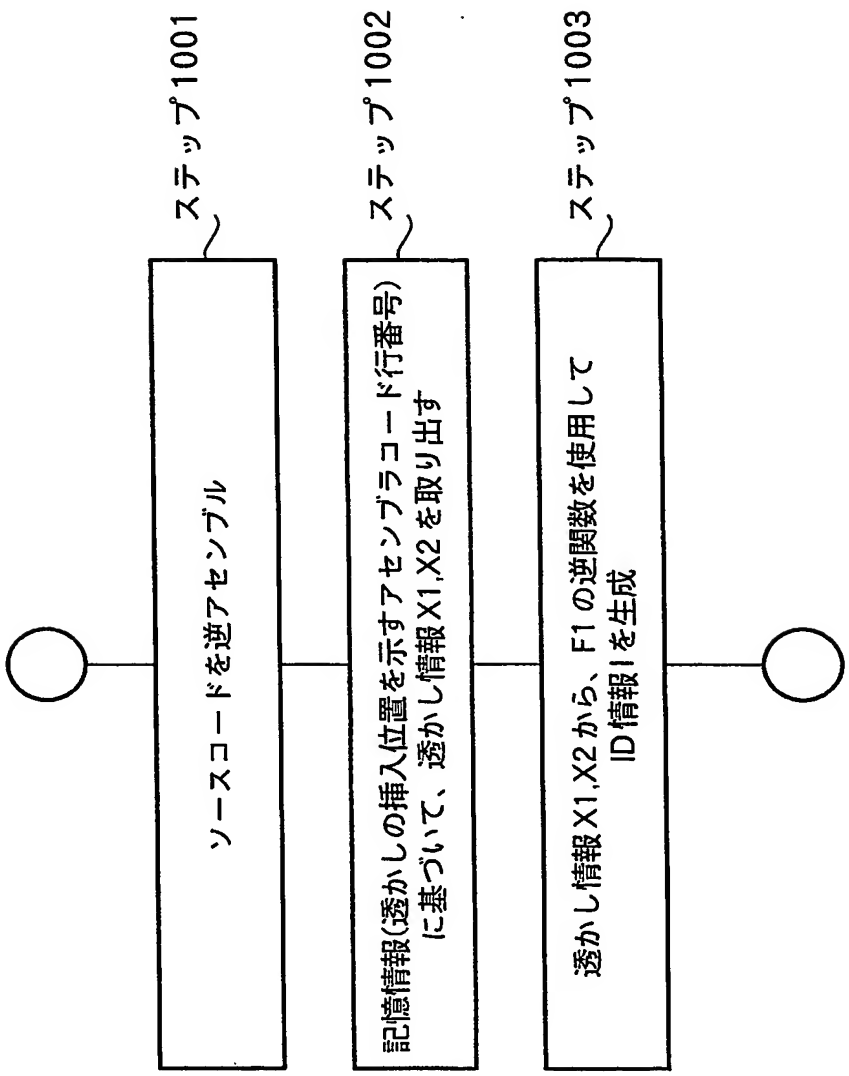
【図４】


ステップ401
F(1)により透かし情報X1,X2を生成
ステップ402
定数C1,C2を出力するF21(X1,X2)、F22(X1,X2)を構成
ステップ403
・"val1 = X1 ; "; "val2 = X2 ; " をコード中に埋め込む
ステップ404
・"val3 = F21(X1,X2)"; "val4 = F22(X1,X2)" を
コード中に埋め込む
ステップ405
"if(val3 ! = C1){assert(0)}"; "if(val4 ! = C2){assert(0)}"を
コード中に埋め込む
ステップ406
透かし情報を挿入した個所を記憶

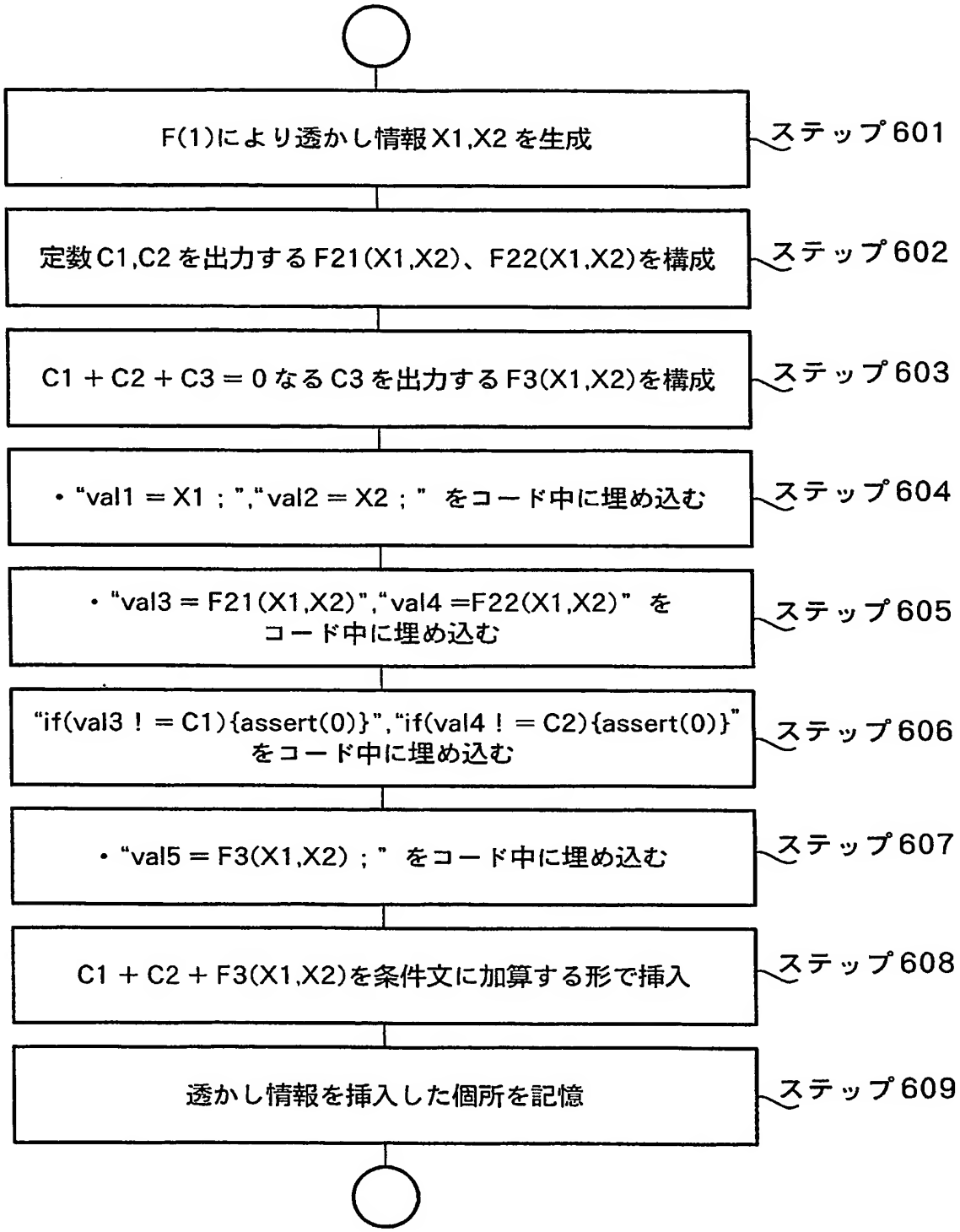
【図５】

500a
gcd(int x,int y){
int a=x,b=y,c;
while(b!=0){
a=a % b;
c=a;a=b;b=c;
if(a<0)a=-a;
}
return a;
}
500b
int val1=1234,val2=5678;
gcd(int x,int y){
int a=x,b=y,c;
int val3=val1+0.195667*val2;
int val4=3.700972*val1+0.195667*val2;
while(b!=0){
a=a % b;
c=a;a=b;b=c;
if(a<0)a=-a;
if(val3!=2345)assert(0);
if(val4!=5678)assert(0);
}
return a;
}
500c
int val1=1111,val2=2222;
gcd(int x,int y){
int a=x,b=y,c;
int val3=val1+0.555356*val2;
int val4=4*val1+0.555356*val2;
while(b!=0){
a=a % b;
c=a;a=b;b=c;
if(a<0)a=-a;
if(val3!=2345)assert(0);
if(val4!=5678)assert(0);
}
return a;
}
501
502
503

【図６】


ステップ1001
ソースコードを逆アセンブル
ステップ1002
記憶情報(透かしの挿入位置を示すアセンブラコード行番号)
に基づいて、透かし情報X1,X2を取り出す
ステップ1003
透かし情報X1,X2から、F1の逆関数を使用して
ID情報Iを生成

【図７】


F(1)により透かし情報X1,X2を生成
ステップ601
定数C1,C2を出力するF21(X1,X2)、F22(X1,X2)を構成
ステップ602
C1＋C2＋C3＝0なるC3を出力するF3(X1,X2)を構成
ステップ603
・“val1＝X1；”,“val2＝X2；”をコード中に埋め込む
ステップ604
・“val3＝F21(X1,X2)”,“val4＝F22(X1,X2)”をコード中に埋め込む
ステップ605
“if(val3！＝C1){assert(0)}”,“if(val4！＝C2){assert(0)}”をコード中に埋め込む
ステップ606
・“val5＝F3(X1,X2)；”をコード中に埋め込む
ステップ607
C1＋C2＋F3(X1,X2)を条件文に加算する形で挿入
ステップ608
透かし情報を挿入した個所を記憶
ステップ609

【図８】

800a

```
gcd(int x,int y){
int a=x,b=y,c;
while(b!=0){
a=a % b;
c=a; a=b; b=c;
if( a<0)a=-a;
}
return a;
}
```

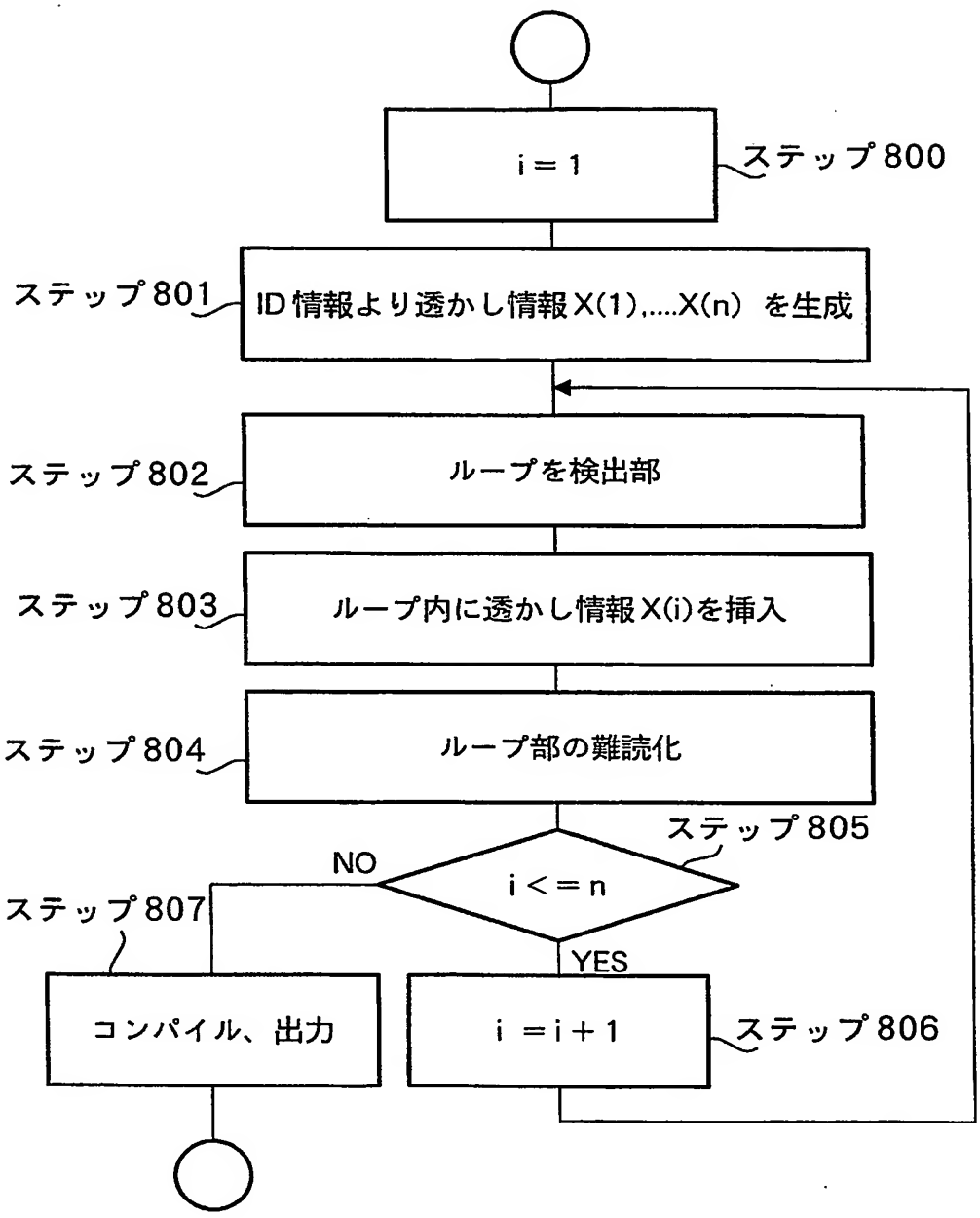
↓

800b

```
int val1=1234,val2=5678;                          ~701

gcd(int x,int y){
int a=x, b=y, c;
int val3=val1+0.195667*val2;
int val4=3.700972*val1+0.195667*val2;             } ~702
int val5=5.601297*val1+0.195667*val2;
while(b!=0){
  a=a % b;
  c=a;a=b;b=c;
  if(a<val3+val4+val5)a=-a;                       ~703
  if(val3!=2345)a++;
  if(val4!=5678)assert(0);                        } ~704
}
return a;
}
```

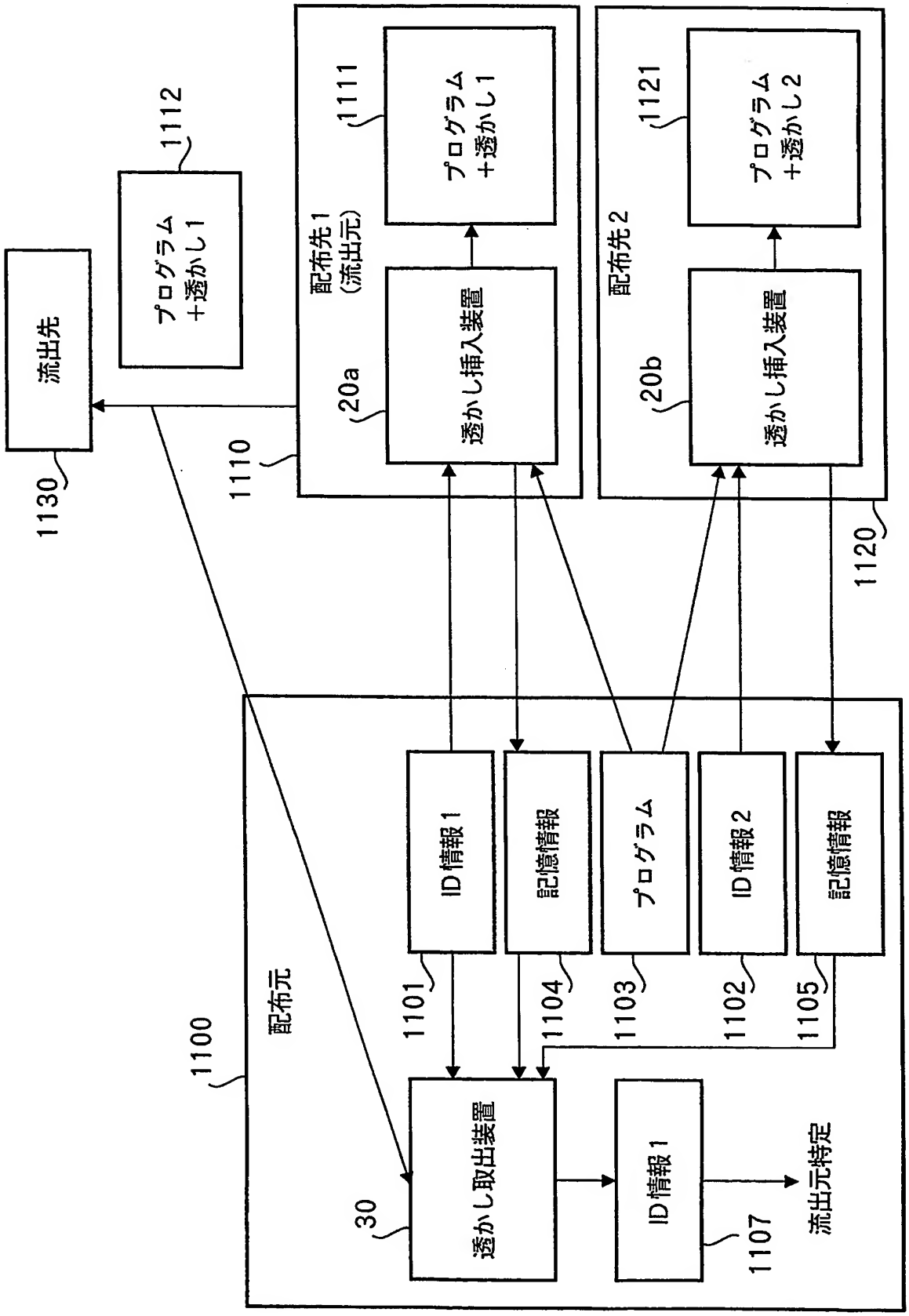
【図９】


i＝1
ステップ800
ステップ801
ID情報より透かし情報X(1),....X(n) を生成
ステップ802
ループを検出部
ステップ803
ループ内に透かし情報X(i)を挿入
ステップ804
ループ部の難読化
ステップ805
NO
i＜＝n
ステップ807
YES
コンパイル、出力
i ＝i＋1
ステップ806

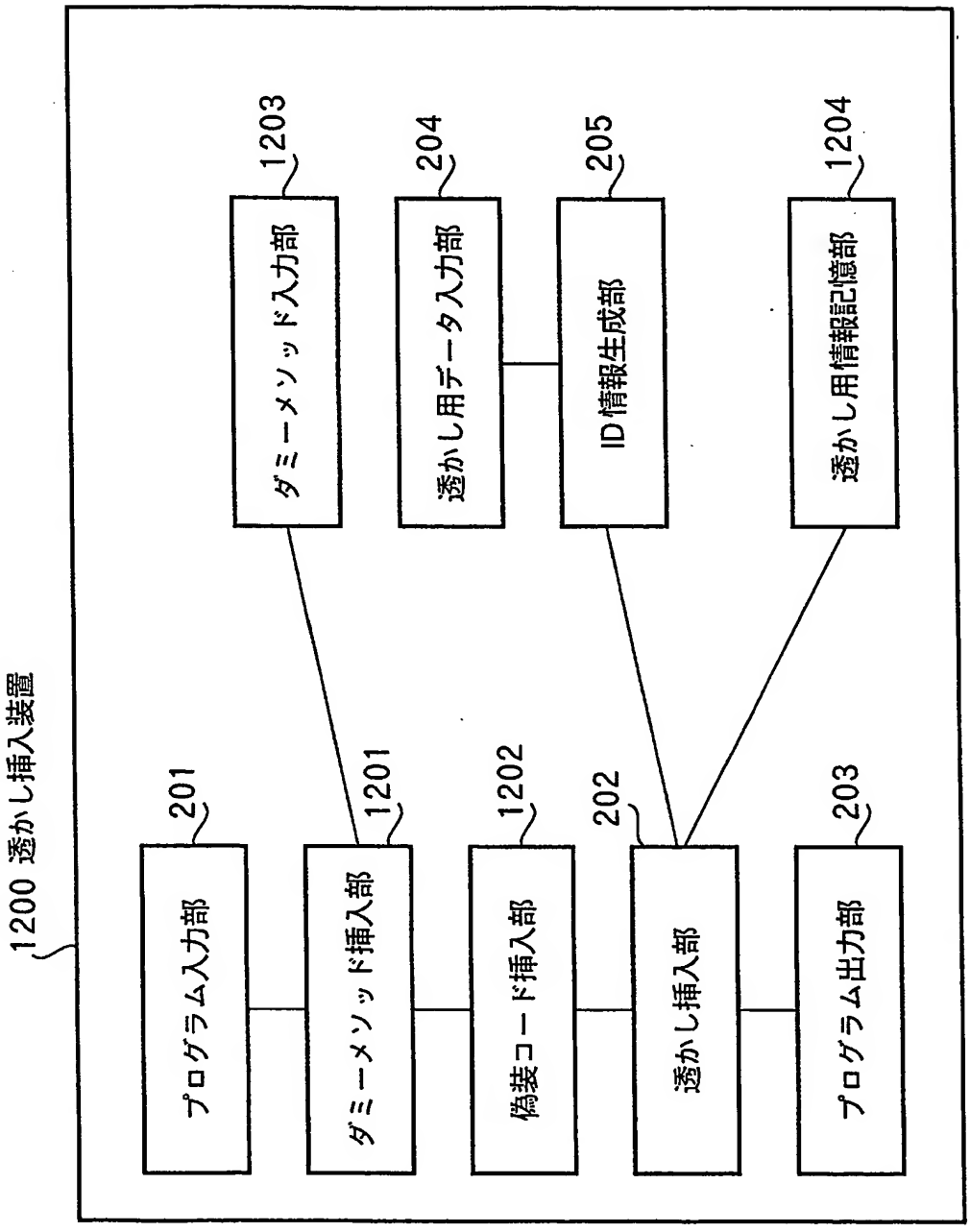
【図１０】

900a
sample(int *A, int N){
int i;
int S=A[0];
for(i =1;i<N;i++){
if(S<A[i])S=A[i]
{
returs S;
}
900b
int val1;
sample(int *A, int N){
int S =A[0];
int i;
while(1){
val1=12345678;
if(i>=N)break;
if(S<A[i])S=A[i];
i++;
if(i>=N)break;
if(S<A[i])S=A[i];
i++;
}
return S;
}
902a
901
900c
int val1;
sample(int *A, int N){
int i=1;
int S =A[0];
while(i<N){
902b
if(i>=N)break;
if(S<A[i])S=A[i];
val1=11112222;
N--;
if(i<N)break;
if(S<A[i])S=A[N];
i++;
}
return S;
}

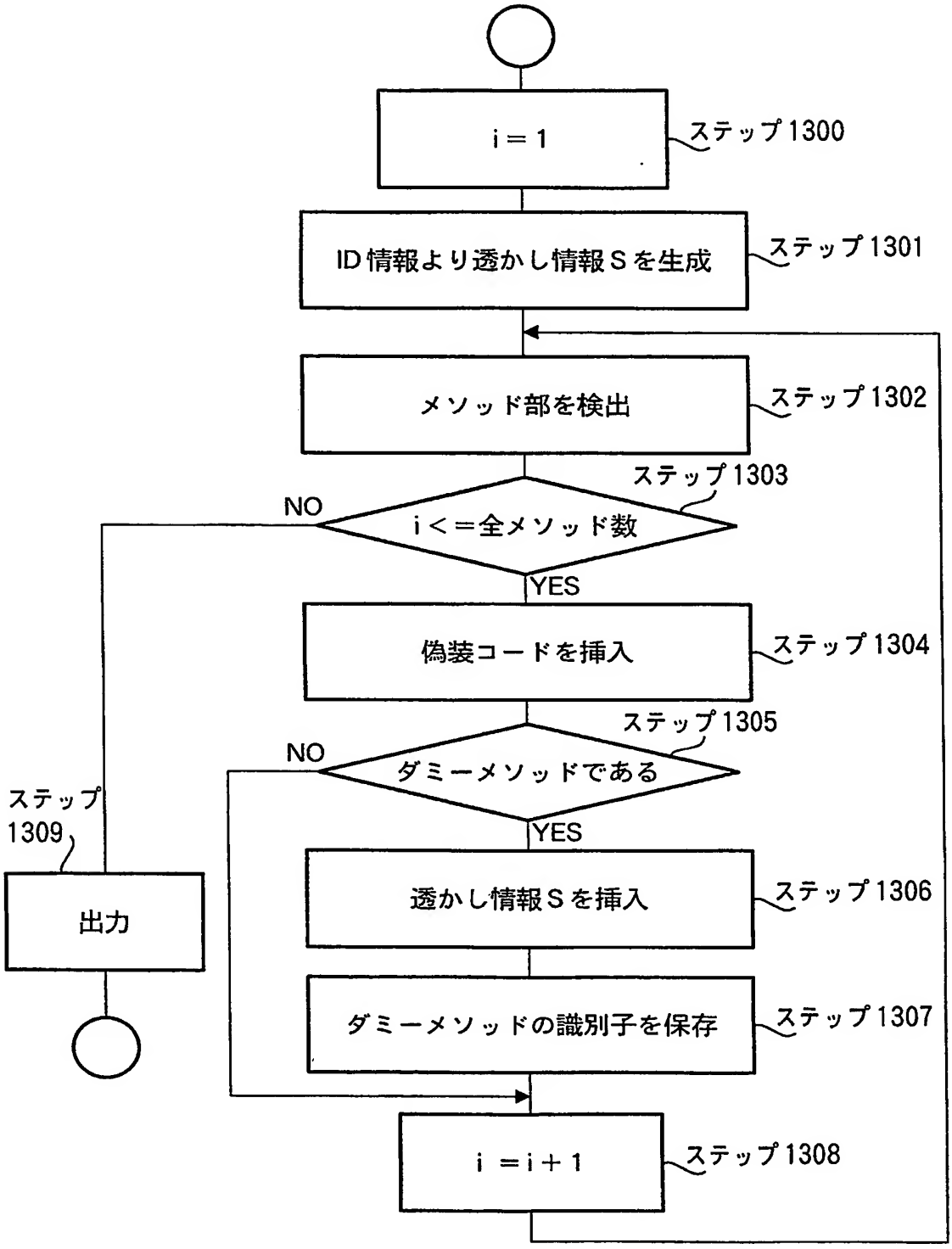
【図１１】


流出先
1130
プログラム
＋透かし１
1112
配布先１
（流出元）
1110
プログラム
＋透かし１
1111
20a
透かし挿入装置
配布先２
1120
プログラム
＋透かし２
1121
20b
透かし挿入装置
配布元
1100
ID情報１
1101
記憶情報
1104
プログラム
1103
ID情報２
1102
記憶情報
1105
30
透かし取出装置
ID情報１
1107
流出元特定

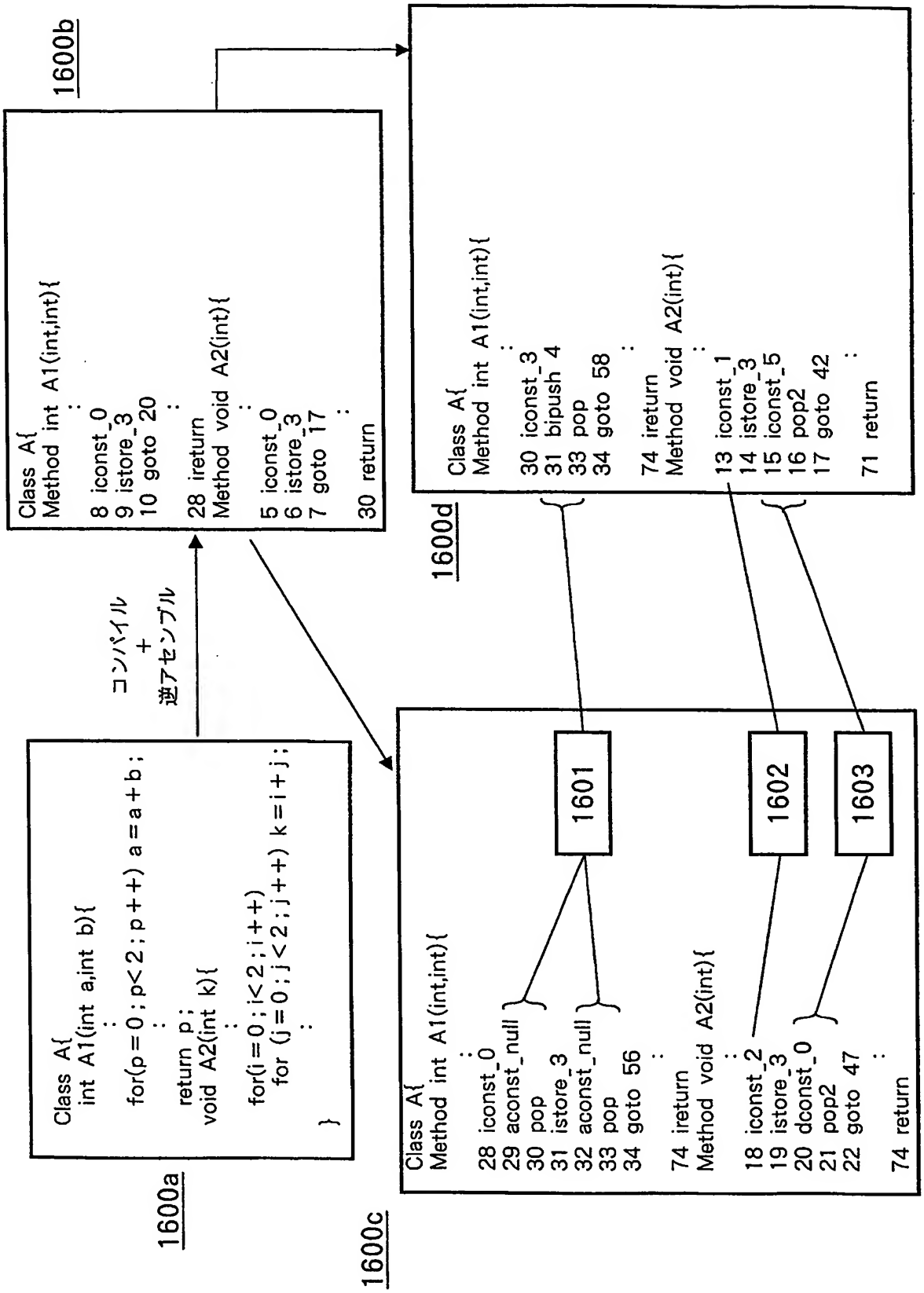
【図１２】


1200 透かし挿入装置
プログラム入力部
201
ダミーメソッド挿入部
1201
偽装コード挿入部
1202
透かし挿入部
202
プログラム出力部
203
ダミーメソッド入力部
1203
透かし用データ入力部
204
ID情報生成部
205
透かし用情報記憶部
1204

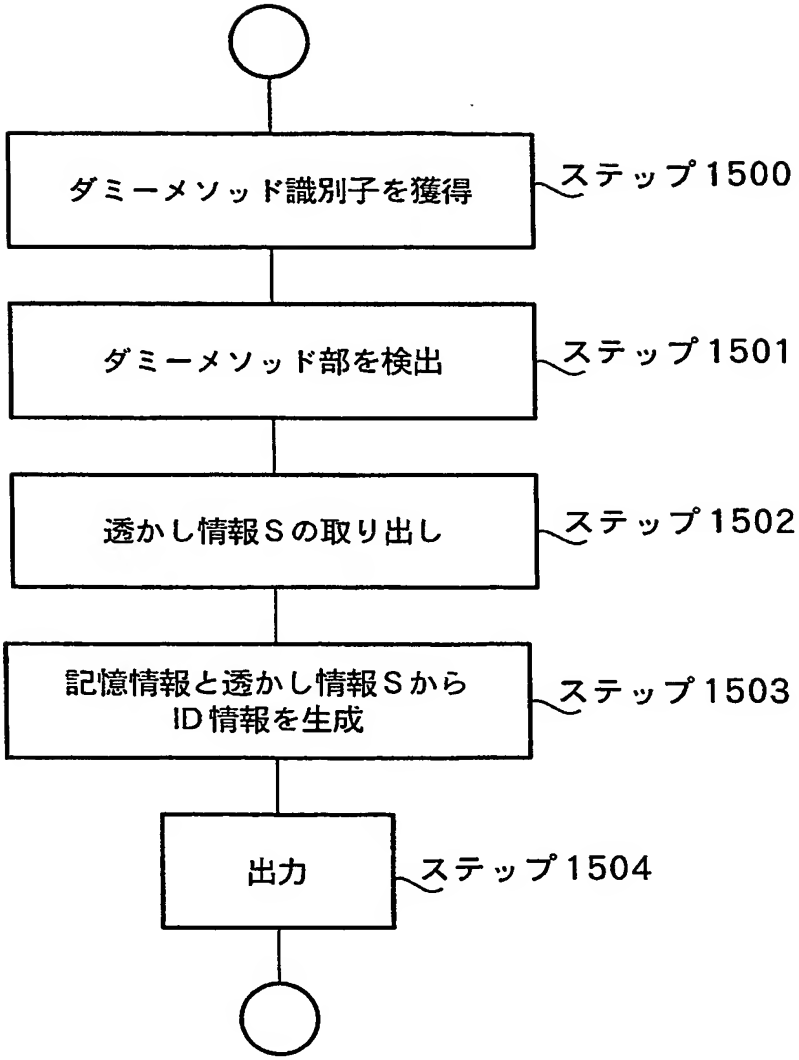
【図１３】


i＝1
ステップ1300
ID情報より透かし情報Ｓを生成
ステップ1301
メソッド部を検出
ステップ1302
ステップ1303
NO
i＜＝全メソッド数
YES
偽装コードを挿入
ステップ1304
ステップ1305
NO
ダミーメソッドである
YES
ステップ
1309
出力
透かし情報Ｓを挿入
ステップ1306
ダミーメソッドの識別子を保存
ステップ1307
i ＝i＋1
ステップ1308

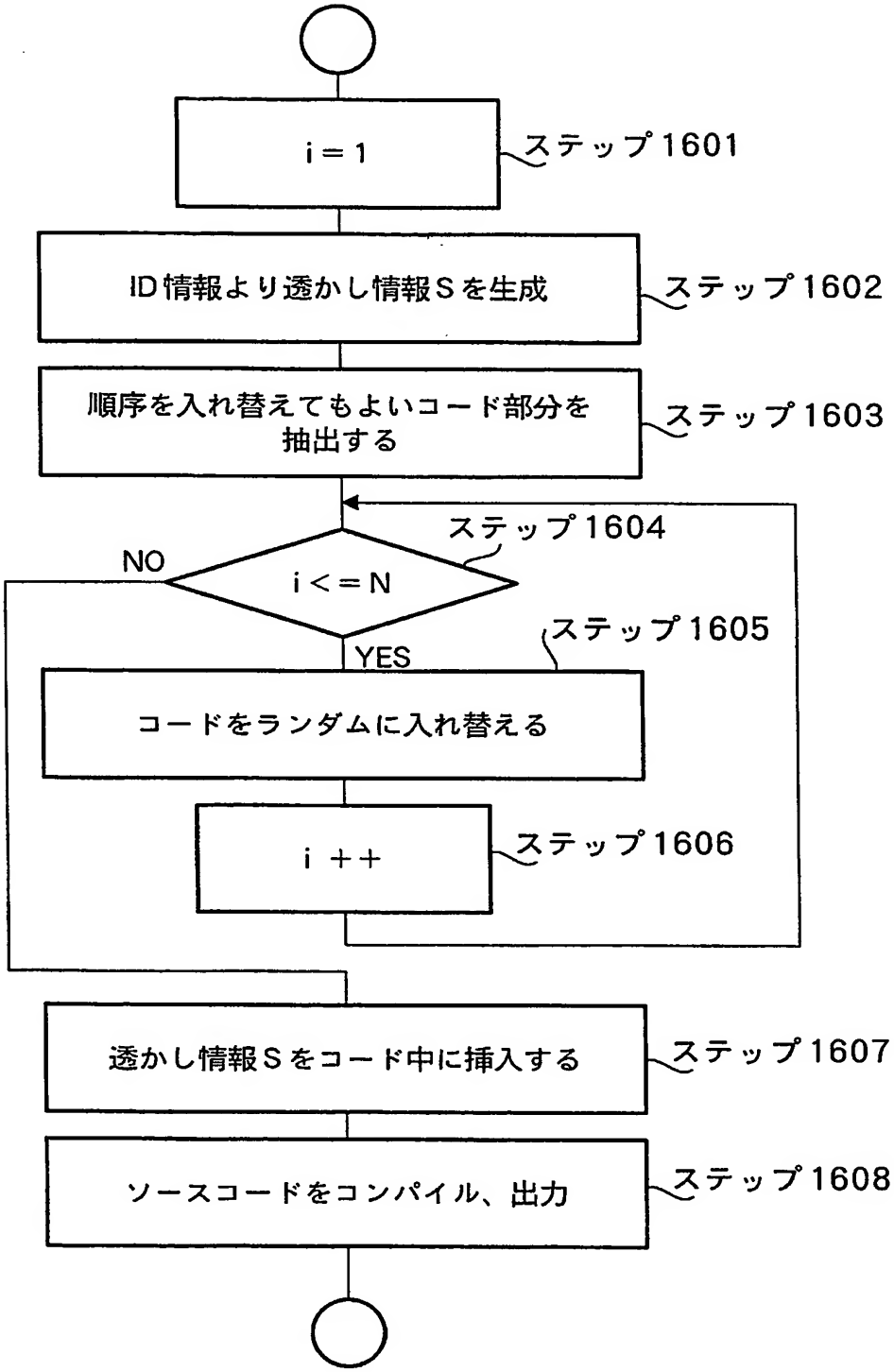
【図１４】


1600a
Class A{
int A1(int a,int b){
:
for(p=0;p<2;p++) a=a+b;
:
return p;
void A2(int k){
:
for(i=0;i<2;i++)
for (j=0;j<2;j++) k=i+j;
:
}
コンパイル
+
逆アセンブル
1600b
Class A{
Method int A1(int,int){
:
8 iconst_0
9 istore_3
10 goto 20
:
28 ireturn
Method void A2(int){
:
5 iconst_0
6 istore_3
7 goto 17
:
30 return
1600d
Class A{
Method int A1(int,int){
:
30 iconst_3
31 bipush 4
33 pop
34 goto 58
:
74 ireturn
Method void A2(int){
:
13 iconst_1
14 istore_3
15 iconst_5
16 pop2
17 goto 42
:
71 return
1600c
Class A{
Method int A1(int,int){
:
28 iconst_0
29 aconst_null
30 pop
31 istore_3
32 aconst_null
33 pop
34 goto 56
:
74 ireturn
Method void A2(int){
:
18 iconst_2
19 istore_3
20 dconst_0
21 pop2
22 goto 47
:
74 return
1601
1602
1603

【図１５】


ダミーメソッド識別子を獲得
ステップ1500
ダミーメソッド部を検出
ステップ1501
透かし情報Ｓの取り出し
ステップ1502
記憶情報と透かし情報Ｓから
ID情報を生成
ステップ1503
出力
ステップ1504

【図１６】


i＝1
ステップ1601
ID情報より透かし情報Ｓを生成
ステップ1602
順序を入れ替えてもよいコード部分を抽出する
ステップ1603
ステップ1604
NO
i＜＝N
ステップ1605
YES
コードをランダムに入れ替える
i ＋＋
ステップ1606
透かし情報Ｓをコード中に挿入する
ステップ1607
ソースコードをコンパイル、出力
ステップ1608

【図１７】

1700a
temp(){
int x=1 ,y=2, z=3;
int i;
int sum=0;
1701a
for(i =0;i<5;i++){
sum=x+y+z;
x++;
y+=2;
z+=3;
printf("sum==%d\n",sum);
1703a
}
}
1700c
temp(){
int i;
int sum=0;
int x=1 ,y=2, z=3;
int a=321
1701c
1702c
for(i =0;i<5;i++){
sum=x+y+z;
x++;
z+=3;
y+=2;
printf("sum==%d\n",sum);
1703c
}
}
1700b
temp(){
int sum=0;
int x=1 ,y=2, z=3;
int i;
1701b
for(i =0;i<5;i++){
int a=123
1702b
sum=x+y+z;
z+=3
y+=2;
x++;
printf("sum==%d\n",sum);
1703b
}
}

【書類名】要約書
【要約】
【課題】　透かしの挿入箇所を特定されないように透かしを挿入することにより、透かし情報がなく、かつ正常に動作するプログラムを生成できないようにすること。
【解決手段】　本発明は、プログラムの配布先を一意に特定するＩＤ情報から透かし情報を生成し、前記透かし情報を前記プログラムに挿入し、前記透かし情報挿入個所周辺や前記プログラム全体を前記プログラムの仕様が変更しないように前記配布先ごとに改変し、前記透かし情報を用いたものであって、前記透かし情報が改ざんされた場合には、前記プログラムを正しく動作させないものであり、前記配布先毎に異ならないコードを前記プログラムに挿入するようにすることにより、差分攻撃により透かしであるコードが検出されないようにした。
【選択図】　図２

特願２００３－３２４８０５

出願人履歴情報

識別番号　［０００００５８２１］

１．変更年月日　１９９０年　８月２８日
［変更理由］　新規登録
住　所　大阪府門真市大字門真１００６番地
氏　名　松下電器産業株式会社